CARVE: BREAKING DATA BARRIERS IN CHIP PLACEMENT BY HARNESSING REUSABLE EXPERTISE

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained macro-placement policies can reduce repeated optimization across circuits, but deployment often exposes them to unfamiliar designs when the original training data are unavailable. Repeatedly fine-tuning a single serving model can overwrite earlier improvements, while simply saving checkpoints does not determine where they can be reliably reused. We introduce Continual Adaptation through the Reuse of Validated Expertise (CARVE), a framework that represents accumulated expertise as a frozen base policy, immutable specialists, and task-specific credentials obtained through local validation. For a new task, CARVE first checks existing specialists and trains a new specialist from the frozen base only when none qualifies. Under fixed task distributions and validation rules that control cumulative error, we establish expected-performance guarantees for repeated reuse. For bounded losses, we also derive matching worst-case bounds on the local samples needed for reliable reuse. In macro placement, a reuse-first follow-up reduces recorded training time by 58.5% (9.66 to 4.01 hours), while mean HPWL gain changes only from 8.41% to 7.86%. In a simulated receiving deployment, imported specialists are reused on six of seven new IBM circuits with no receiver-side training, achieving a 5.76% mean HPWL gain. Navigation studies provide complementary evidence on repair retention and repeated adaptation.

1 INTRODUCTION

Macro placement in hardware design determines where large circuit blocks are positioned in an application-specific integrated circuit (ASIC), affecting wirelength and routing congestion. Conventional placement uses commercial electronic design automation tools with input from human experts and can require substantial time and effort (Lee et al., 2025). In contrast, learning-based approaches can reduce this effort by using circuit data and feedback on placement quality to train policies that select macro locations. For example, Mirhoseini et al. (2021) report generating a placement for a TPU block in six hours after pretraining, compared with several weeks for their human expert baseline. The learned placement also achieved shorter wirelength. However, realizing these efficiency gains in deployment depends on whether an organization can adapt and reuse the policy with the design data available to it.

Access to design data constrains how learned placement policies can be reused. Circuit netlists and placement histories can contain proprietary information, and licensing restrictions limit the data available for learning (Chai et al., 2022). Access to a pretrained placement policy does not necessarily include access to its source training data. For example, AlphaChip (Goldie et al., 2024) provides a pretrained checkpoint that other researchers have fine-tuned on their own circuit benchmarks, despite incomplete availability of the original training data (Cheng et al., 2026). We consider deployments in which an organization receives such a policy but cannot access the provider's training circuits or placement histories. The organization can evaluate and adapt the policy on its own circuits, but joint retraining or replay requiring unavailable provider examples is not possible. Adaptation using only new-task data nevertheless remains possible (Li & Hoiem, 2017). Within this data constraint, deployed policies also face an out-of-distribution (OOD) challenge as new circuits arrive. Differences in macro sizes, netlist connectivity, and placement constraints may produce tasks poorly represented in the training data. Such differences do not necessarily prevent a learned policy from producing a good placement (Lai et al., 2023), but performance on earlier circuits does not establish whether it will perform well on a new design. An operator therefore needs to evaluate the avail-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

able policy on each new circuit and determine whether further adaptation is needed. Methods such as ChiPFormer (Lai et al., 2023) support adaptation through additional placement trials and policy updates. However, their pretraining and fine-tuning procedure does not specify how to retain the resulting adaptations, validate them for later circuits, or select among saved models as requests arrive. The deployment problem therefore extends beyond adapting to an individual circuit to accumulating local improvements and reusing suitable models without the provider's training data.

Together, restricted access to source data and unfamiliar circuits motivate a continual-learning-based approach that adapts a pretrained policy using local data while retaining earlier capabilities. Specifically, placement trials on local circuits should support adaptation to unfamiliar designs (Lai et al., 2023), while the approach should also retain the resulting improvements for later reuse. However, adaptation alone leaves several deployment requirements unresolved. For instance, EWC (Kirkpatrick et al., 2017) and Synaptic Intelligence (Zenke et al., 2017) use parameter importance statistics from earlier learning that may not accompany pretrained weights. Methods that freeze earlier models (Rusu et al., 2016) can preserve their behavior, but freezing alone does not establish whether those models can serve a new task without further training. In addition, successively updating the shared model can change parameters needed by earlier tasks, causing catastrophic forgetting (Kirkpatrick et al., 2017). Although replay-based approaches (Lopez-Paz & Ranzato, 2017) can mitigate forgetting by revisiting earlier examples, provider training data are unavailable in our setting and cannot be used for replay. Another possible approach would be Learning without Forgetting, which can operate without earlier training examples by matching the previous model's predictions (Li & Hoiem, 2017) on current-task inputs. However, placement states from a new circuit may poorly represent conditions encountered on earlier circuits, leaving earlier behavior insufficiently protected during adaptation. Finally, naively extended training can lead to a higher risk of loss of plasticity (Dohare et al., 2024), which impedes subsequent learning even when new data are available (Lyle et al., 2023). Neuron recycling, parameter resets, and regularization can mitigate this decline using local data, but their effectiveness depends on the network and training conditions. An intervention targeting one mechanism may leave other causes unresolved (Dohare et al., 2024; Lyle et al., 2025). A reliable solution therefore requires preserving accumulated expertise, establishing where it can be reused, and supporting further learning from local data.

Motivated by the above, we introduce Continual Adaptation through the Reuse of Validated Expertise (CARVE), a framework that accumulates adapted policies as specialists using a pretrained base and local task data. We introduce a key concept called a specialist, which is a copy of the pretrained policy adapted to a task using local data. Each specialist is stored with task-specific authorizations, called credentials, that record where it has passed evaluation against the base. To retain earlier capabilities, CARVE keeps the base and accepted specialists unchanged, so training for a new task cannot overwrite an earlier specialist. When an earlier task returns, its authorized specialist remains available without reconstructing its training data. To avoid carrying accumulated repair updates into later learning, each new specialist is trained from a separate copy of the same frozen base. Its initial parameters therefore remain independent of the sequence of earlier repairs, preventing successive adaptations from progressively changing the starting point for later learning. Training and validation use current-task data without requiring provider examples, previous-task replay, or historical parameter importance statistics.

To decide whether to reuse an existing specialist or train a new one, CARVE first evaluates its saved specialists and the frozen base on the same local cases for the current task. A specialist that passes a predeclared paired comparison receives a credential for that task under the chosen metric. The credential authorizes subsequent use without changing the specialist, allowing one model to serve several tasks after separate evaluations. If no examined specialist qualifies, CARVE trains a new candidate from the frozen base and accepts it only after it passes the same comparison. Each request is served before the repair it triggers, so new specialists and credentials affect later requests. Each task receives at most one bounded training session per repair epoch, and the base serves whenever no specialist is authorized. An imported specialist follows the same validation procedure at the receiving operator, where reuse requires local evaluation without access to its source circuits or training histories. CARVE therefore combines unchanged models with task-specific evidence for deciding whether to reuse existing expertise or train a new specialist. Our contributions are:

- **A continual-adaptation abstraction based on validated expertise.** CARVE represents learned capability as immutable specialists with task-specific authorizations, separating

capability acquisition, retention, reuse, and deployment decisions. It checks saved specialists and starts bounded local training only when none qualifies, without requiring provider training data or previous-task replay.

- **A statistical characterization of reliable reuse.** We establish conditions for preserving expected performance under repeated reuse and matching worst-case sample bounds under bounded losses for a fixed library of specialists.
- **Empirical evidence across macro placement and navigation.** We study repeated repair, cross-task reuse, task-local model selection, and simulated cross-operator transfer, quantifying both the optimization savings and the quality trade-offs that arise when previously learned expertise is reused instead of training a task-specific specialist.

2 CONTINUAL ADAPTATION WITH CARVE

CARVE is our approach for supporting continual adaptation in macro placement without provider training data, shown in Figure 1. Its library stores the frozen base, unchanged specialists, and task approvals. Each request uses an approved specialist or the base. When repair is triggered, local evaluation can authorize saved specialists for later requests. If none qualifies, training starts from the base, and only passing candidates enter the library. Imported specialists undergo the same local checks, enabling reuse across operators with current-task data alone.

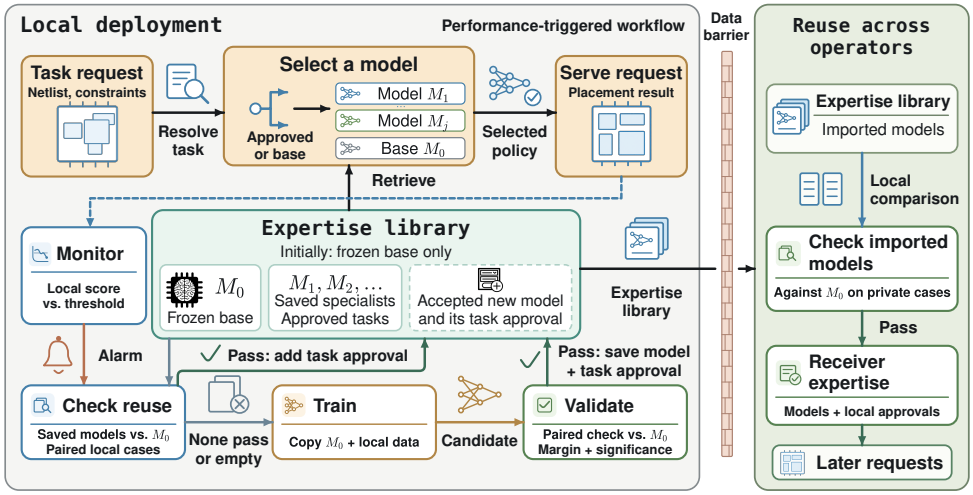


Figure 1: Overview pipeline of CARVE.

2.1 RETAINED MODELS AND PERMITTED USES

We consider a request x_t that asks a policy to solve a task at arrival t. For placement, it supplies a circuit's netlist, macro sizes, and placement constraints, and the policy returns macro locations. Each request may contain several trials or episodes, each called a case z. A family f groups repeated requests for the same task, and we denote $\mathcal{F}$ as the set that contains all the declared families. Each placement family is one circuit with its macro and netlist specifications. The deterministic resolver ρ assigns $f_t = \rho(x_t)$, with $\perp$ denoting an unresolved or out-of-scope request. A repair epoch fixes the base policy M_0, resolver, metric, and contract $\mathcal{C}$, which specifies the trigger, trainer, budget, evaluation rule, and model selector. We denote the expertise library as $\mathcal{M}_t = \{M_0, \ldots, M_{J_t}\}$ that contains the frozen base M_0 and J_t unchanged specialists, each a separately adapted policy. The credential set Γ_t records approvals (j, f) permitting M_j to serve family f under the epoch's base, metric, and contract. Local evaluation for issuing a credential is called admission. The fixed selector $\sigma(f, \Gamma_t)$ chooses among approved specialists, and g_t denotes the serving model. Initially, the library contains only M_0 and has no credentials. The set $\mathcal{T}_t$ records families whose new candidate received a complete rejection. The set $\mathcal{X}_t$ records families whose entire training session exhausted its allowed attempts without producing a candidate. Eligible families are resolved families with no credential and no stopped status. Monitoring cases are denoted by E_t, and local admission cases by D_f^{dev}.

The score $s_f(M, z)$ evaluates model M on case z and is oriented so that larger values are better. Placement uses an equivalent rule favoring lower half-perimeter wirelength (HPWL), an estimate of wiring length.

Given x_t, the resolver supplies the family key used to retrieve an approved specialist. The selector returns g_t, which produces the requested output before any resulting repair. Unresolved or stopped families, and families without approval, would still use M_0. Returning tasks can therefore reuse an unchanged specialist without reconstructing earlier training data or repeating validation. Since selection is restricted to evaluated pairs (j, f), approval on one circuit does not authorize service on another. After the base service for an eligible family, the monitor applies $\mathcal{C}$ to E_t and returns a repair decision. A trigger then starts an extension exam, which compares saved specialists with M_0 on the same local cases using the admission rule in Section 2.2. The contract fixes the comparison set, order, and multiplicity correction. A pass adds (j, f) to Γ_t without changing weights, extending earlier expertise to another family. Training begins when no examined specialist passes or the comparison set is empty. Newly issued credentials affect only later requests. The main placement study instead trains each new circuit before logging extensions, which neither avoid training nor change routing. The follow-up evaluates checking before training. The receiver takes imported specialists and evaluates them against its own base on private target cases. A pass creates a local credential for later service, using the same extension procedure. Only the models need to cross the data barrier, while source circuits and training histories remain unnecessary. Source credentials cannot replace local evaluation, and the receiver study performs no training and uses its base if none passes.

2.2 Training and admitting a new repair

During training, for family f, let D_f^{tr} denote local training data, Train_f the trainer with fixed settings ξ_f, and C_f its candidate policy. A session permits at most r training attempts. Admission compares C_f with M_0 on D_f^{dev} using observed improvement $\widehat{\Delta}_f$ and required margin m_f. When a test is used, p_f denotes its p-value and α_f its declared threshold. Let P_f be the fixed distribution of family f's cases, including task settings and execution randomness. A fixed loss $\ell_f(M, z) \in [0, 1]$ defines expected loss $R_f(M) = \mathbb{E}_{Z\sim P_f}[\ell_f(M, Z)]$ and expected improvement $\Delta_f(M) = R_f(M_0) - R_f(M)$. The score $s_f = 1 - \ell_f$ connects this analysis to paired evaluation. Let k index validation rounds, $\mathcal{G}_{k-1}$ denote their prior history, and α_k their error allowance chosen from that history. The event B_k occurs when round k issues any credential with $\Delta_f(M_j) < 0$; δ is the total error tolerance.

For one target family, fix $J \geq 1$ specialists before drawing n independent cases $Z_1, \ldots, Z_n$ from P_f. Write $\Delta_j = \Delta_f(M_j)$, $\Delta_0 = 0$, and $\widehat{\Delta}_j$ for the sample mean paired improvement. A validator selects $\widehat{j} \in \{0, \ldots, J\}$, where zero denotes the base. Reliable reuse requires nonnegative selected improvement and selection of a specialist whenever some specialist improves by at least $\gamma > 0$, with probability at least $1 - \delta$. Let $n^\star$ be the smallest fixed sample size meeting this requirement uniformly over all libraries and target distributions with bounded losses. The population threshold γ differs from the empirical admission margin m_f.

Train. Training computes $C_f = \mathsf{Train}_f(M_0, D_f^{\mathrm{tr}}; \xi_f)$ when the reuse check finds no passing specialist. Each candidate starts from a separate copy of the base, so earlier repairs leave its initial parameters and all saved models unchanged. Each family receives at most one bounded session per epoch. Retries are permitted only when no candidate is produced, never after rejection. Training uses current-family data without provider examples or replay from previous families.

Validate and retain. The candidate and base receive the same admission cases, and their scores determine whether the candidate passes the declared rule. A contract using both a margin and a test requires $\widehat{\Delta}_f \geq m_f$ and $p_f \leq \alpha_f$, and a margin-only contract omits the test, while the margins were selected during development. The protocol requires separate training and admission data; the navigation records do not establish disjointness. A pass saves C_f as M_{J_t+1} together with credential $(J_t + 1, f)$ and its evaluation evidence before later service. Rejection adds f to $\mathcal{T}_t$, while training exhaustion adds it to $\mathcal{X}_t$. Both retain the base for the epoch. Incomplete evidence authorizes neither service nor a performance rejection. Admission thus separates learning a candidate from permitting its use, while preservation retains each accepted repair. The following results explain how unchanged specialists support repeated reuse and when local evaluation can authorize reuse before training.

Proposition 1 (Expected performance under repeated reuse)**.** *Suppose the base, stored specialists, inference procedures, losses, and family distributions remain fixed within an epoch. Start with no credentials and issue each through validation. For* $0 < \delta < 1$*, assume* $\Pr(B_k \mid \mathcal{G}_{k-1}) \le \alpha_k$ *and* $\sum_{k\ge1} \alpha_k \le \delta$ *almost surely. If* g_t *is the base or a specialist credentialed for* f_t*, then*

$$\Pr[R_{f_t}(g_t) \le R_{f_t}(M_0) \text{ for every in-scope request } t] \ge 1 - \delta. \tag{1}$$

Proof sketch. A conditional union bound gives $\Pr(\bigcup_k B_k) \le \mathbb{E}[\sum_k \alpha_k] \le \delta$. Otherwise every credential has nonnegative expected improvement. Unchanged models and distributions preserve that comparison, and selection uses the corresponding credential. Fallback gives equality. Appendix E.1 gives the proof. New credentials consume the error budget; repeated use does not. The guarantee concerns expected loss, allowing variation across cases and between successive specialists. An unchanged family key alone does not ensure an unchanged distribution.

Local evaluation for reliable reuse. The requirement defined above is

$$\Pr\left[\Delta_{\widehat{j}} \ge 0 \text{ and } \left(\max_{1\le j\le J} \Delta_j \ge \gamma \implies \widehat{j} \ne 0\right)\right] \ge 1 - \delta. \tag{2}$$

The validator learns the target distribution only through the paired losses on these cases; the library and n are fixed beforehand.

Theorem 2 (Local evaluation for reliable reuse)**.** *For* $J \ge 1$ *and* $0 < \gamma, \delta \le 1/4$*, we have*

$$\frac{\log(J/\delta)}{64\gamma^2} \le n^\star \le \left\lceil \frac{8\log(2J/\delta)}{\gamma^2} \right\rceil. \tag{3}$$

Thus $n^\star = \Theta(\gamma^{-2}\log(J/\delta))$*, no source training data or parameter updates needed for upper bound.*

Proof sketch. The paired estimate is $\widehat{\Delta}_j = n^{-1}\sum_{i=1}^n [\ell_f(M_0, Z_i) - \ell_f(M_j, Z_i)]$. Authorize a specialist when $\widehat{\Delta}_j \ge \sqrt{2\log(2J/\delta)/n}$. Hoeffding's inequality and a union bound control all paired estimates simultaneously. At the stated sample size, every authorized specialist has nonnegative expected improvement and every specialist improving by at least γ passes, with probability at least $1 - \delta$. Appendix E.2 shows the lower bound follows by distinguishing unknown target distributions that make different fixed specialists useful from a distribution where all specialists are harmful.

Theorem 2 explains how local evidence can authorize saved or imported expertise before training. It counts cases, while evaluation can require $(J + 1)n$ executions. The selected specialist need not be best, improve by γ, or match a newly trained model. Both results require representative fresh validation and error control across credential decisions. The experimental navigation and median-HPWL gates do not directly instantiate the bounded-loss rule. Appendix E.3 gives a concrete rule and its scope.

3 Evaluation design

We first use crowd navigation as a controlled numerical study of CARVE, followed by our main chip-placement evaluation. Both domains involve a pretrained policy, recurring tasks, and local feedback for evaluating repairs. Navigation lets us vary scene conditions and request order explicitly to examine retention, admission, and reuse. Within each domain, methods share a pretrained base and the same recorded request order, but maintain separate repair states. Requests are served before repair, so updates affect later requests. Appendix A.6 gives the domain-specific rules.

Prior work evaluates candidates using an empirical promotion threshold (Silver et al., 2017) or an explicit performance requirement with statistical error control (Thomas et al., 2015). We selected CARVE's domain margins from development experiments. Navigation requires an increase of at least 15 percentage points in swept success on 100 paired cases. Placement requires at least a 5% relative reduction in median HPWL on 30 paired trials, together with the declared paired test. The margins specify the minimum observed improvement required for admission. The code is available at https://anonymous.4open.science/r/CARVE-610B/.

3.1 Crowd navigation

Navigation instantiates Section 2 with scene families in place of circuits. A family f specifies geometry and the pedestrian model, while a request x_t specifies scene size and pedestrian count. A

case z also fixes episode randomness. The frozen base M_0 is the pretrained SARL policy (Chen et al., 2019) in SICNav (Samavi et al., 2025). New candidates use only current-family simulations. Credentials (j, f) authorize unchanged specialists for a family, retaining the same selection, reuse, and fallback rules.

Each run contains 30 requests of 100 matched cases. Success requires reaching the goal within 90 seconds without swept-volume contact. Measured success triggers repair, and saved specialists are checked before training. Figure 2 shows the scene families and the first requests in the fixed-arrival sequence.

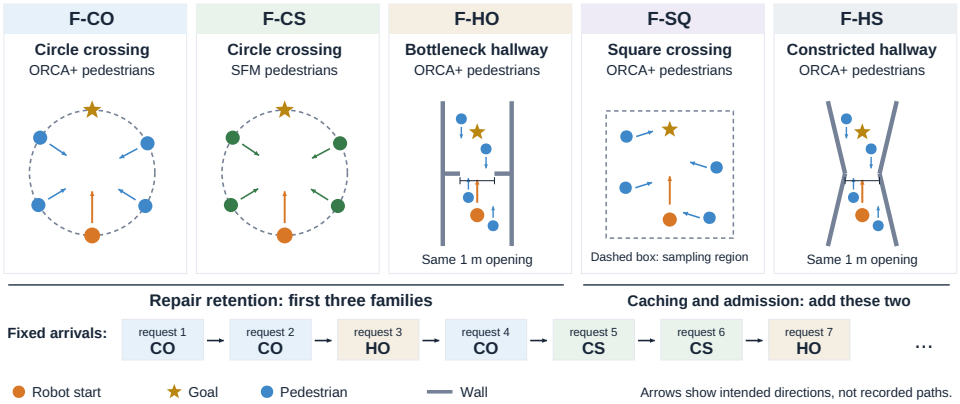


Figure 2: Navigation scene families and the fixed-arrival request stream.

The repair retention study uses circle crossing with ORCA+ or SFM pedestrians (Van Den Berg et al., 2011; Helbing & Molnar, 1995; Zhang et al., 2026) and an ORCA+ bottleneck hallway. Frozen keeps the base unchanged, Overwrite repeatedly fine-tunes and replaces one active policy, and CARVE preserves accepted specialists. Appendix A.5 gives the complete rules. Fixed arrivals follow a predefined request list, while sampled arrivals draw requests from the same pool of scene settings. The caching and admission study adds square crossing and a constricted hallway. It samples request order and scene settings while fixing the number of requests from each family. Cache-only retains a model for each family without an admission check. Adapted CompoNet (Malagon et al., 2024) provides a modular continual-learning comparison on separate case blocks with the same arrival order and three root seeds. Neither method was evaluated on the fixed-arrival or sampled-arrival repair retention streams. Complete workload and comparison settings are in Appendix A.1.

Repair retention. CARVE reduces repeated training and has higher success than Overwrite under both arrival constructions in Table 1. Table 3 shows that, with fixed arrivals, it saves separate circle and hallway specialists, retrieves the circle model when its task returns, and extends it to SFM without retraining. Repeated hallway repairs account for most of Overwrite's cost, but CARVE's hallway success stays near the base. Repair time includes training and all admission and extension checks, including failures, but excludes ordinary service.

Table 1: Repair retention with fixed and sampled arrivals.

	Fixed arrivals					Sampled arrivals				
Method	Success (%)	Train count	Train (h)	Checks (h)	Repair (h)	Success (%)	Train count	Train (h)	Checks (h)	Repair (h)
Frozen	35.5	0	0.00	0.00	0.00	34.3	0	0.00	0.00	0.00
Overwrite	59.1	21	13.09	0.00	13.09	70.5	22	10.84	0.00	10.84
CARVE	**67.3**	2	1.39	0.34	**1.73**	**71.9**	2	1.30	0.29	**1.60**

Caching and admission. CARVE has higher success and lower checkpoint storage than Cache-only on the displayed stream, while Cache-only has lower repair time in Table 2. Checks add evaluation work, and the additional arrival stream gives a different quality comparison. Cache-only uses 25% more storage under the same six-entry limit. Extra storage is $100(B/B_{\mathrm{CARVE}}-1)$, using unrounded registered-file sizes. Storage measures checkpoint files, not RAM or GPU memory.

Table 2: Caching and admission with fixed family counts.

Method	Success (%)	Train count	Train (h)	Checks (h)	Repair (h)	Storage files / MiB
Frozen	40.9	0	0.00	0.00	0.00	—
Overwrite	57.9	25	14.16	0.00	14.16	—
CompoNet	54.4	5	2.49	0.00	2.49	—
Cache-only	57.9	5	2.85	0.00	**2.85**	5 / 1.872
CARVE	**59.8**	3	2.23	0.73	2.96	**4 / 1.498**

Table 3: Fixed arrivals, requests 1 to 15 of 30. Success is out of 100 cases; repair cost includes training and checks after each request.

Request	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Family	**CO**	**CO**	**HO**	**CO**	**CS**	**CS**	**HO**	**CO**	**CO**	**CS**	**HO**	**CO**	**CO**	**HO**	**CS**
Size $r[w]$	2	2.25	1.75	2	1.375	1.375	1.75	1.5	1.25	2.25	1.75	2.5	1.375	1.75	1.375
Pedestrians	5	4	5	5	4	5	5	5	2	4	4	3	5	5	4
Frozen success	20	11	27	22	80	74	26	35	51	36	39	7	46	28	78
Overwrite success	20	97	18	20	96	91	28	69	89	92	26	88	45	13	94
CARVE success	20	97	27	81	80	96	28	64	94	95	40	92	65	29	98
Overwrite cost (min)	20.2	0.0	63.0	20.8	0.0	0.0	64.5	21.0	0.0	0.0	55.3	14.8	21.2	65.4	0.0
CARVE cost (min)	26.2	0.0	76.4	0.0	1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 3 shows the circle specialist serving later circle requests after hallway adaptation and receiving approval for SFM without retraining. Navigation therefore tests the retention and reuse decisions that the placement studies next examine on circuit workloads.

3.2 CHIP MACRO PLACEMENT

Our main application tests whether CARVE can retain and reuse local placement repairs without the provider's training data. We use the released ChiPFormer checkpoint (Lai et al., 2023) and adapt it using only the current circuit, without using the provider's training dataset or placement histories. The experiments follow how a local repair is retained when its circuit returns, reused for a different circuit, and then used by a simulated receiving operator. We measure whether each form of reuse improves placement with less new optimization, while respecting the source-data restriction.

Adapting locally and retaining repairs. We first test whether local adaptation can improve later placement without repeating the optimization whenever a circuit returns. The circuit revisit study contains 24 requests over eight circuits, each appearing three times. Three circuits are in the reported pretraining corpus and five are not. Each request provides macro sizes, connections, and a placement region; the policy assigns macro positions. Figure 3 illustrates the placement task and the arrival sequence, including a circuit returning after other circuits have been processed.

Frozen, Overwrite, RS, Cache-only, and CARVE start from the same checkpoint and process the same order with 30 matched service seeds per request. Overwrite updates one active policy; Cache-only saves a model for each circuit; RS checks candidates but uses one active model across circuits. CARVE retrieves a saved model for a circuit on which it has passed evaluation. Each request is served before any resulting repair, so a new model can improve only later requests. In the main study, CARVE gives each new circuit its own training session. Extensions recorded afterward do not skip training; the next experiment tests that use of extension. Appendix A.6 gives the full deployment rules. Half-perimeter wirelength (HPWL) estimates wirelength. Gain is the percentage reduction in a request's median HPWL relative to Frozen. Mean gain and its standard deviation summarize the 24 requests; mean rank averages each method's placement in the per-request HPWL ranking. Repair time includes training and all checks, including failures, but excludes ordinary service.

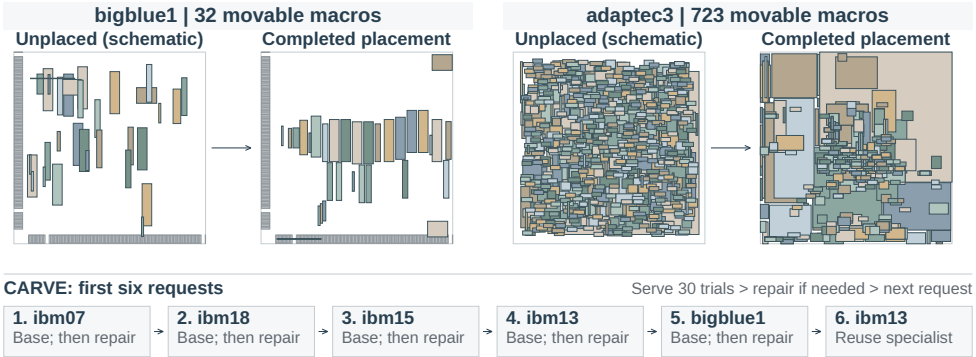


Figure 3: Macro-placement examples and the first six requests in the circuit revisit study. Scattered layouts are schematic; completed layouts are archived frozen-base outputs, not CARVE repair results.

Keeping a repair useful when its circuit returns. CARVE obtains the highest mean HPWL gain and lowest mean rank among the five methods on the recorded stream, with less repair work than Overwrite and RS (Table 4). Cache-only also avoids repeated training and is close in aggregate quality and repair cost, so the benefit of preserving models is shared by both methods. The comparison therefore concerns how retained models are used as well as whether they are stored.

Table 4: Placement quality and repair time over 24 requests.

Method	HPWL gain (%) mean ± SD	Mean rank	Train (h)	Checks (min)	Repair (h)
Frozen	0.00 ± 0.00	2.917	0.00	0.00	0.00
Overwrite	−7.63 ± 13.72	3.917	19.56	0.00	19.56
RS	0.39 ± 10.48	2.792	31.93	24.58	32.34
Cache-only	7.04 ± 9.03	1.917	10.53	0.00	10.53
CARVE	8.41 ± 10.05	1.750	9.66	42.26	10.36

Table 5 connects the aggregate result to that use. After repairing `ibm13`, CARVE processes other circuits and retrieves the earlier `ibm13` specialist on both revisits, without new training. The rejected `adaptec3` candidate incurs repair cost but adds no serving model, and the next `ibm13` request still uses its saved repair. Local learning can thus accumulate useful repairs without recovering source training data or overwriting earlier models. The request records show earlier optimization benefiting later service, without another training job for each return.

Table 5: Circuit revisits, requests 1 to 12. Repair cost includes training and checks; storage follows each request. Complete records and timing boundaries are in Appendix A.3.

Request	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Circuit	ibm07	ibm18	ibm15	ibm13	bigblue1	ibm13	ibm18	bigblue1	ibm07	ibm10	adaptec3	ibm13
Arrival	1	1	1	1	1	2	2	2	2	1	1	3
HPWL gain over Frozen (%)												
Overwrite	0.0	−0.6	0.8	4.5	4.8	−7.0	−11.5	3.4	−10.3	8.0	−37.0	4.9
RS	0.0	0.0	0.0	8.2	6.5	0.4	−20.7	9.3	−7.6	8.0	−15.4	3.9
Cache-only	0.0	0.0	0.0	0.0	0.0	17.0	−2.8	9.2	11.2	0.0	0.0	9.9
CARVE	0.0	0.0	0.0	0.0	0.0	20.7	17.6	9.5	16.3	0.0	0.0	16.0
Repair cost and CARVE storage												
Overwrite cost (min)	89.8	98.4	80.1	80.1	18.9	80.1	80.1	19.3	82.0	80.1	3.4	80.1
CARVE cost (min)	90.0	85.5	85.4	85.5	23.9	0.0	0.0	0.0	0.0	85.6	83.1	0.0
Storage (MiB)	37.3	74.6	112.0	149.3	186.6	186.6	186.6	186.6	186.6	223.9	223.9	223.9

Reusing a repair for a new circuit. To make accumulated expertise useful on new designs, we test whether a saved repair can replace further training without replaying earlier circuit data. We compare training for each new circuit with checking saved specialists first, using the same request order and service seeds. The base serves each circuit's first request before any check or training, and a passing specialist serves later requests unchanged.

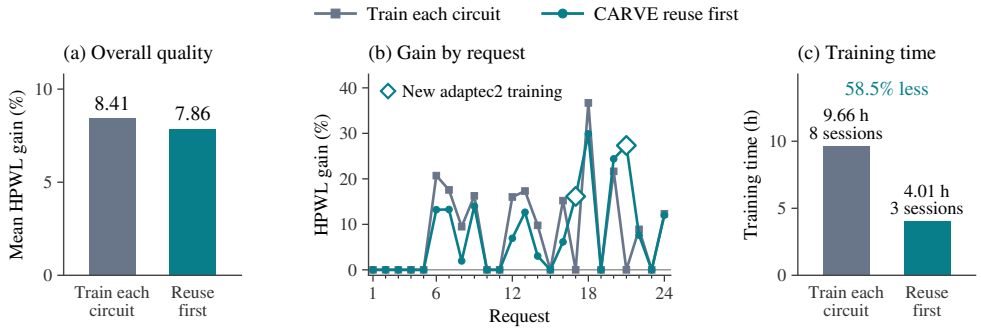


Figure 4: Reuse across 24 circuit requests. Gain is the median HPWL reduction versus Frozen on 30 matched trials.

Figure 4 shows that local validation extends one `ibm07` repair to five new circuits, reducing training work. On the five reused circuits, reuse improves over Frozen but gives lower mean gains than separate training; new `adaptec2` training partly offsets the difference overall. The result supports reusing accumulated expertise, with a quality trade-off. Different training seeds and integration code limit causal attribution, as shown in Appendix B.2.

Reusing another operator's models without its data. To test whether an existing library helps an operator without its source data, a simulated operator B imports six specialists from operator A. It checks them on seven new IBM circuits and an `adaptec1` control using 30 paired trials. The earliest source-admitted passing model serves on 30 separate seeds, with Frozen as fallback. Figure 5 shows checking the full library costs 20.71 minutes across eight targets, with no receiver training. Overall, CARVE achieves a 5.76% mean HPWL gain over Frozen across all eight targets, averaging each target's median gain and including both base fallbacks. Six new circuits use imports, showing useful reuse without source training data. The 419.17-minute source training cost was paid once. Per-target results, uncertainty, and timing definitions remain in Appendix B.3.

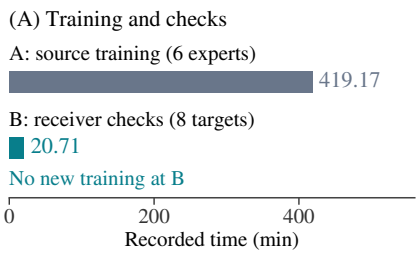


Figure 5: Training and checks.

Selecting a model checked for the arriving circuit. Saved repairs need evidence of suitability for the arriving circuit. In four consecutive requests, RS reuses its `bigblue1` specialist, while CARVE retrieves each circuit's validated repair, as shown in Table 6. Gains are similar on `bigblue1`. On `ibm18` and `ibm07`, RS worsens HPWL while CARVE improves it. Different models and training histories prevent isolating the check's causal effect. Full HPWL values, model identities, and the complete RS trace, including beneficial uses across circuits, remain in Appendix C.4. The chip studies show how retained models support reuse when circuits return, on new circuits, and across operators. Local evaluation extends their use without source design data and can reduce new training.

Table 6: Median HPWL gain over Frozen (%) on 30 matched service trials. Higher is better.

Request	Circuit	RS	CARVE
6	`ibm13`	0.39	**20.71**
7	`ibm18`	-20.73	**17.57**
8	`bigblue1`	9.33	**9.48**
9	`ibm07`	-7.61	**16.27**

4 Related Works

Continual World evaluates retention and transfer during sequential reinforcement learning (Wołczyk et al., 2021). Progressive Neural Networks preserve earlier columns (Rusu et al., 2016), PackNet allocates parameters through pruning (Mallya & Lazebnik, 2018), and CompoNet retains and combines policy modules (Malagon et al., 2024). CARVE uses model isolation and studies the deployment decisions around saved models. The experiments compare whether a candidate is checked, whether its task scope is recorded, and whether the appropriate specialist is used when a task returns. Cache-only and RS test those choices with the same domain trainer. Learning without Forgetting uses new-task data to retain earlier behavior (Li & Hoiem, 2017), while Tent adapts selected parameters using unlabeled test inputs (Wang et al., 2020). Source-data restrictions therefore do not rule out all existing adaptation methods. CARVE studies episodic tasks with local evaluation and separates generating a candidate from allowing it to serve a particular family. The proposed plan and base-penalty studies examine that distinction for navigation. High Confidence Policy Improvement provides probabilistic guarantees under its stated assumptions (Thomas et al., 2015). ChiPFormer supplies the pretrained placement model and adaptation procedure (Lai et al., 2023). Our placement studies compare deployment rules around that procedure, including repeated use, extension to new circuits, and local evaluation of imported models.

5 Conclusion

CARVE keeps a repair as a separate model and records the tasks on which it has passed evaluation. In the evaluated navigation and placement streams, retaining specialists reduces repeated training and preserves the earlier model identities. Service quality is less uniform, with useful rejections and unfavorable navigation results alongside gains. Local extension tests allow some new tasks to reuse a saved model without training, although reuse can improve quality less than training for the target. The component studies show why retention, admission, and task scope should be evaluated separately. The resulting evidence concerns an automated repair procedure with measured costs and outcomes, rather than a guarantee that every repair or transfer improves future service.

Reproducibility statement

We provide the code in the supplementary material, together with a README file that allows experimental results to be reproduced.

The use of LLM

We used LLM to polish the writing and correct grammar in some paragraphs, but it did not contribute to ideas or conceptual content.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. volume 34, pp. 29304–29320, 2021.

Gavin C Cawley and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research*, 11:2079–2107, 2010.

Zhuomin Chai, Yuxiang Zhao, Yibo Lin, Wei Liu, Runsheng Wang, and Ru Huang. Circuitnet: An open-source dataset for machine learning applications in electronic design automation (eda). *Science China Information Sciences*, 65(12):227401, 2022.

Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *2019 international conference on robotics and automation (ICRA)*, pp. 6015–6022. IEEE, 2019.

Chung-Kuan Cheng, Andrew B. Kahng, Sayak Kundu, Yucheng Wang, and Zhiang Wang. An updated assessment of reinforcement learning for macro placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 45(8):3654–3668, 2026. doi: 10.1109/TCAD.2025.3644293.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632 (8026):768–774, 2024.

Anna Goldie, Azalia Mirhoseini, and Jeff Dean. That chip has sailed: A critique of unfounded skepticism around ai for chip design, 2024.

Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51 (5):4282, 1995.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Yao Lai, Jinxin Liu, Zhentao Tang, Bin Wang, Jianye Hao, and Ping Luo. Chipformer: Transferable chip placement via offline decision transformer. In *International Conference on Machine Learning*, pp. 18346–18364. PMLR, 2023.

Vint Lee, Minh Nguyen, Leena Elzeiny, Chun Deng, Pieter Abbeel, and John Wawrzynek. Chip placement with diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 33499–33514. PMLR, 13–19 Jul 2025.

Zhizhong Li and Derek Hoiem. Learning without forgetting. volume 40, pp. 2935–2947. IEEE, 2017.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International conference on machine learning*, pp. 23190–23211. PMLR, 2023.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, Hado van Hasselt, Razvan Pascanu, James Martens, and Will Dabney. Disentangling the causes of plasticity loss in neural networks. In Vincenzo Lomonaco, Stefano Melacci, Tinne Tuytelaars, Sarath Chandar, and Razvan Pascanu (eds.), *Proceedings of The 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 750–783. PMLR, 2025.

Mikel Malagon, Josu Ceberio, and Jose A. Lozano. Self-composing policies for scalable continual reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34432–34460. PMLR, 21–27 Jul 2024.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773. IEEE, 2018.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nova, et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Sepehr Samavi, Anthony Lem, Fumiaki Sato, Sirui Chen, Qiao Gu, Keijiro Yano, Angela P Schoellig, and Florian Shkurti. Sicnav-diffusion: Safe and interactive crowd navigation with diffusion trajectory predictions. *IEEE Robotics and Automation Letters*, 10(9):8738–8745, 2025.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *International Conference on Machine Learning*, pp. 2380–2388. PMLR, 2015.

Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics research: the 14th international symposium ISRR*, pp. 3–19. 2011.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. 2020.

Maciej Wołczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. volume 34, pp. 28496–28510, 2021.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995. PMLR, 2017.

Jiefu Zhang, Yang Xu, and Vaneet Aggarwal. Don't freeze, don't crash: Extending the safe operating range of neural navigation in dense crowds, 2026. URL https://arxiv.org/abs/2603.06729.

A MAIN EXPERIMENT RECORDS

This appendix reports benchmark workloads, service records, and repair costs. Reuse and transfer appear in Appendix B, and component studies appear in Appendix C. Measurement definitions are in Appendix D. Methods, domain rules, and terms are collected in Appendices A.5, A.6, and A.7. B1 denotes Frozen, B3 denotes Cache-only, and OV denotes Overwrite. Archived chip slots start at zero, while displayed request numbers start at one.

A.1 NAVIGATION WORKLOADS AND REPAIR WORK

The *repair retention study* compares Frozen, Overwrite, and CARVE on circle crossing with ORCA+ (CO) or SFM (CS) pedestrians and an ORCA+ bottleneck hallway (HO). Fixed and sampled arrivals use the same pool of scene settings. The *caching and admission study* adds Cache-only, square crossing (SQ), and a constricted hallway (HS). It samples request order and scene settings while fixing the number of requests per family. We show one such stream in detail and report the additional completed stream below.

Each stream starts from the same frozen base with an empty repair state. The method set and workload differ between the two studies, so their results do not isolate admission alone. Table A1 gives the family counts in the three detailed streams.

Table A1: Workload composition of the three detailed streams. Each request evaluates 100 matched cases.

			Repair retention		Caching and admission
Geometry	Pedestrians	Family	Fixed arrivals	Sampled arrivals	Fixed family counts
Circle crossing	ORCA+	F-CO	14	16	8
Circle crossing	SFM	F-CS	7	7	5
Bottleneck hallway	ORCA+	F-HO	9	7	6
Square crossing	ORCA+	F-SQ	—	—	6
Constricted hallway	ORCA+	F-HS	—	—	5
Request positions			30	30	30
Original-batch methods			3	3	4
Original-batch service episodes			9,000	9,000	12,000

Scope of the caching and admission comparison. The two completed streams use the same family counts but different sampled orders and scene settings. The detailed display was chosen during editing without a predeclared exclusion rule. In the other completed stream, CARVE achieves 57.2% success, compared with 60.7% for Cache-only and 62.9% for Overwrite. CARVE trains fewer times than both methods in each stream, but its success advantage does not hold across both streams. Its total repair time is lower than Overwrite's and higher than Cache-only's in both streams. The full records remain in the archive.

Fixed and sampled arrivals for repair retention. Fixed arrivals follow the predefined list of 30 requests. Sampled arrivals draw 30 requests from 43 allowed scene settings, comprising 25 CO, nine CS, and nine HO settings. Settings can repeat, but the sampler redraws a setting if it has already appeared four times. Sampling changes both the family counts and the arrival order (Table A1). Each request evaluates 100 cases, and repeated settings use the next consecutive block of 100 cases.

Sampled arrivals for caching and admission. Both streams contain eight CO, five CS, six HO, six SQ, and five HS requests. The generator first shuffles this family list, then draws scene settings for each request. SQ uses widths 3, 4, and 5 with three to five pedestrians. HS uses the recorded radius field 1.25, 1.5, or 1.75 with three to five pedestrians. A setting is redrawn if it has no eligible case block.

For each setting, the generator excludes blocks already used in the repair retention study or the SQ/HS census. It takes the next eligible 100-case block in ascending order without reusing a block within that stream. Admission and service use different case blocks; Appendix C.2 describes their relationship to training. A case is defined by its family, scene settings, and index, so the same index with different settings does not identify the same case.

Reproducing the arrival histories. The archived request lists retain each stream's order, scene settings, and generator configuration. Arrival randomization is separate from model-training randomness.

Circle and square outlines mark sampling regions, not walls; square pedestrians start and finish on opposite sides. Both hallway openings are 1 m within a 2 m corridor, but the bottleneck uses a transverse wall and HS uses sloping walls. The recorded `circle_radius` is not the opening width. The square shim repairs missing environment references without adding walls.

Table A2: Fixed navigation execution settings. Case allocation is described in Appendix C.2.

Setting	Value
Base and simulator	Frozen SARL, archived CrowdSimPlus
Service and episode limit	100 episodes per request, 90 seconds per episode
Alarm	Two-sided 95% Clopper–Pearson lower endpoint below .80; equivalently at most 88 successes
Local training	60,000 full fine-tuning steps, learning rate 2.5×10^{-5}, batch size 64
Training data	Current family only; no replay from earlier families
Admission and extension	Same 100 cases for candidate and base; at least 15 additional swept successes
Observed service blocks	100 matched cases per request; separate from admission
Cache limit	Four entries in repair retention; six entries for both cached methods in caching and admission

Across fixed arrivals, sampled arrivals, and caching and admission, Overwrite performs 3, 3, and 5 first-family updates and 18, 19, and 20 later updates, for totals of 21, 22, and 25. CARVE runs 2, 2, and 3 training sessions, passes 1, 1, and 2 extension exams, and records 27, 21, and 20 cache hits, respectively. Cache-only trains five models in the caching and admission stream. Later Overwrite updates are not individually classified as parameter forgetting.

The three detailed streams have 11 first and 57 subsequent Overwrite updates. Exact CARVE phase totals are in Table D1.

Retaining models reduces repeated work. Cache-only has lower total repair cost in the displayed caching and admission stream. With fixed arrivals, hallway repairs dominate Overwrite's cost, while CARVE's hallway success stays close to the base. SFM reuse avoids training but still incurs an extension check.

Table A3: Family breakdown for fixed arrivals over all 30 requests. Repair includes training and all checks. Frozen has no repair work.

		Success (%)			Trainings		Repair (min)	
Family	Requests	Frozen	Overwrite	CARVE	Overwrite	CARVE	Overwrite	CARVE
Circle, ORCA+	14	27.6	65.6	77.1	11	1	212.9	26.2
Circle, SFM	7	57.9	92.6	93.4	1	0	18.4	1.5
Bottleneck hallway	9	30.4	23.0	31.7	9	1	553.9	76.4

Complete request records. Tables A4, A5, and A6 give one column per request in arrival order, in two panels of 15 requests. Success is out of 100 for every method. Repair minutes are charged after that request, so they affect only later service. A model is named for its training family, not its current authorized target. The records preserve first arrivals, rejections, and unfavorable service. The size row gives radius (width for SQ), and the pedestrian row gives the pedestrian count. After-service actions are T+ (train/admit), T- (train/reject), E+ (extend), or keep (no new permission). The original archive, which also retains the additional arrival, preserves exact blocks, model identities, phase times, and decisions in `all_420_method_requests.csv`; `all_120_requests.csv` has one workload row per request.

Table A4: Fixed arrivals: all 30 requests, with each panel following arrival order. Success is out of 100 cases; repair time is charged after service.

Request	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Family	**CO**	**CO**	**HO**	**CO**	**CS**	**CS**	**HO**	**CO**	**CO**	**CS**	**HO**	**CO**	**CO**	**HO**	**CS**
Size $r[w]$	2	2.25	1.75	2	1.375	1.375	1.75	1.5	1.25	2.25	1.75	2.5	1.375	1.75	1.375
Pedestrians	5	4	5	5	4	5	5	5	2	4	4	3	5	5	4
Frozen success	20	11	27	22	80	74	26	35	51	36	39	7	46	28	78
Overwrite success	20	97	18	20	96	91	28	69	89	92	26	88	45	13	94
CARVE success	20	97	27	81	80	96	28	64	94	95	40	92	65	29	98
CARVE route	base	CO	base	CO	base	CO	HO	CO	CO	CO	HO	CO	CO	HO	CO
After service	T+	keep	T+	keep	E+	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep
Overwrite repair (min)	20.2	0.0	63.0	20.8	0.0	0.0	64.5	21.0	0.0	0.0	55.3	14.8	21.2	65.4	0.0
CARVE repair (min)	26.2	0.0	76.4	0.0	1.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Request	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Family	**CO**	**HO**	**CO**	**CS**	**HO**	**CO**	**CO**	**HO**	**CS**	**CO**	**HO**	**CO**	**CS**	**HO**	**CO**
Size $r[w]$	2	1.25	1.25	2.25	1.75	1.75	1.5	1.75	1.375	2.25	1.75	1.25	2.25	1.25	2.5
Pedestrians	5	5	5	5	4	2	5	5	5	2	4	4	2	5	5
Frozen success	22	24	36	33	38	37	29	24	76	15	40	51	28	28	4
Overwrite success	74	27	52	87	31	93	60	18	94	80	28	60	94	18	71
CARVE success	81	32	65	93	47	96	73	26	97	93	34	67	95	22	92
CARVE route	CO	HO	CO	CO	HO	CO	CO	HO	CO	CO	HO	CO	CO	HO	CO
After service	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep
Overwrite repair (min)	20.6	64.0	21.1	18.4	55.4	0.0	21.2	64.6	0.0	12.8	54.5	19.1	0.0	67.2	20.1
CARVE repair (min)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table A5: Sampled arrivals: all 30 requests, with each panel following arrival order. Success is out of 100 cases; repair time is charged after service.

Request	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Family	**CO**	**CO**	**CS**	**CO**	**CS**	**HO**	**HO**	**HO**	**CO**	**HO**	**CS**	**CS**	**CO**	**CS**	**CO**
Size $r[w]$	2.5	1.5	2.25	1.5	1.375	1.25	1.75	1.5	2	1.25	2.25	1.375	1.375	2.25	2
Pedestrians	2	3	4	2	5	5	5	5	3	5	2	2	4	5	4
Frozen success	13	47	36	41	74	24	27	25	13	28	28	67	45	33	20
Overwrite success	13	85	82	94	90	15	41	36	81	22	69	100	78	95	91
CARVE success	13	85	36	98	92	24	27	25	92	28	94	97	70	91	84
CARVE route	base	CO	base	CO	CO	base	base	base	CO	base	CO	CO	CO	CO	CO
After service	T+	keep	E+	keep	keep	T-	keep	keep	keep	keep	keep	keep	keep	keep	keep
Overwrite repair (min)	13.5	15.4	16.2	0.0	0.0	65.4	63.5	67.5	14.7	66.7	12.6	0.0	18.6	0.0	0.0
CARVE repair (min)	18.5	0.0	5.0	0.0	0.0	72.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Request	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Family	**CO**	**CO**	**CO**	**CO**	**CO**	**HO**	**HO**	**CO**	**CS**	**HO**	**CO**	**CO**	**CO**	**CS**	**CO**
Size $r[w]$	1.25	2.25	1.25	2.5	2	1.5	1.5	2	1.75	1.5	1.25	1.5	1.75	2.25	2
Pedestrians	2	2	2	2	3	5	4	2	4	3	2	2	4	4	4
Frozen success	51	15	57	11	18	19	30	19	43	44	63	55	30	35	17
Overwrite success	87	88	86	83	89	16	42	57	87	44	89	87	87	96	86
CARVE success	87	97	86	97	95	19	30	99	92	44	94	93	83	97	89
CARVE route	CO	CO	CO	CO	CO	base	base	CO	CO	base	CO	CO	CO	CO	CO
After service	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep
Overwrite repair (min)	13.9	12.8	13.7	12.9	0.0	66.4	55.1	13.1	16.4	43.9	0.0	13.1	17.7	0.0	17.4
CARVE repair (min)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table A6: Caching and admission: all 30 requests, with each panel following arrival order. Success is out of 100 cases; repair time is charged after service.

Request	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Family	**CS**	**CS**	**CO**	**CO**	**CS**	**CO**	**HO**	**HS**	**SQ**	**HO**	**HO**	**CO**	**CO**	**CO**	**HS**
Size $r[w]$	2.25	1.375	2.5	2	1.75	1.5	1.25	1.75	4	1.75	1.25	1.75	1.25	1.25	1.75
Pedestrians	5	2	3	4	5	3	5	4	5	3	4	2	3	3	4
Frozen success	33	64	11	22	60	48	23	31	33	51	42	33	55	59	34
Overwrite success	33	100	95	91	96	87	21	22	36	28	37	72	82	88	21
Cache-only success	33	100	11	87	96	86	23	31	33	2	57	87	86	72	52
CARVE success	33	100	11	91	96	87	23	31	33	51	42	98	86	86	52
CARVE route	base	CS	base	CS	CS	CS	base	base	base	base	base	CS	CS	CS	HS
After service	T+	keep	E+	keep	keep	keep	T-	T+	E+	keep	keep	keep	keep	keep	keep
Overwrite repair (min)	18.8	0.0	0.0	0.0	0.0	15.7	68.3	51.8	20.9	45.0	57.7	13.2	16.4	16.6	52.1
Cache-only repair (min)	18.7	0.0	16.0	0.0	0.0	0.0	65.1	50.7	20.8	0.0	0.0	0.0	0.0	0.0	0.0
CARVE repair (min)	23.9	0.0	5.9	0.0	0.0	0.0	77.5	66.8	3.7	0.0	0.0	0.0	0.0	0.0	0.0

Request	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Family	**CO**	**SQ**	**HO**	**HS**	**HO**	**SQ**	**HS**	**CS**	**CO**	**HS**	**HO**	**SQ**	**SQ**	**CS**	**SQ**
Size $r[w]$	1.25	3	1.25	1.75	1.25	5	1.25	2.25	1.5	1.5	1.5	4	5	1.375	3
Pedestrians	5	5	4	4	3	3	3	4	5	4	5	4	3	4	4
Frozen success	47	32	37	41	39	37	57	35	38	39	26	41	44	72	44
Overwrite success	45	44	22	42	41	68	56	79	58	27	33	74	85	97	58
Cache-only success	57	43	43	54	47	77	53	98	62	44	19	54	83	96	51
CARVE success	55	42	37	54	39	75	53	98	68	44	26	59	77	96	52
CARVE route	CS	CS	base	HS	base	CS	HS	CS	CS	HS	base	CS	CS	CS	CS
After service	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep	keep
Overwrite repair (min)	22.1	20.8	58.5	51.8	48.5	14.8	44.3	16.4	22.1	52.8	69.1	18.2	15.3	0.0	18.3
Cache-only repair (min)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CARVE repair (min)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

A.2 Complete fixed-case forgetting sequence

To distinguish shared-parameter forgetting from changing request difficulty, one policy is evaluated on the same fixed circle A and hallway B case blocks after each A–B–A update. Both 100-case blocks lie within their respective training ranges; B training uses no A replay. Circle swept success falls after B training and recovers after retraining A. The historical strict score is retained separately, not substituted for swept success.

Table A7: The same 100 cases per family after each update. Swept success is primary; strict success is a historical diagnostic.

Stage	A swept	B swept	A strict	B strict
Base	16	29	19	70
After learning A	80	15	100	32
After learning B	22	30	23	92
After learning A again	75	27	99	57

The paired A comparison after learning B contains 61 losses and three gains, giving exact McNemar $p = 4.74 \times 10^{-15}$ (McNemar, 1947). This describes the fixed cases, not independent training chains. CARVE retention is checked separately using later serving-model identities; no constant CARVE score curve is inferred.

A.3 Chip request order, work, and stored specialists

E2 follows eight circuits through three arrivals each. The reported pretraining corpus includes adaptec2, adaptec3, and bigblue1; the other circuits are ibm07, ibm10, ibm13, ibm15, and ibm18. Five methods process 30 matched trials per request (3,600 service trials). A request supplies macro sizes, connections, and a placement region. Repair uses only the current circuit, with at most three attempts per session, a 4,800-second budget per attempt, and checkpoints every 600 seconds.

E2 serves before repairing a new family. Later requests retrieve an admitted specialist; rejection consumes repair work without adding a serving file. Post-admission extensions are logged but do

not avoid E2 training. The separate E2-X study instead tests reuse first. Admission and extension rules are stated once below; quality and timing definitions are centralized in Appendix D.

Recorded admission and extension rule. A candidate needs at least 5% lower median HPWL and a one-sided paired signed-rank $p < 0.05$ on 30 matched trials. For extension, the implementation sorts comparisons by p, applies successive Holm thresholds, and stops when either the current threshold or the effect requirement fails. The set contains eight targets per E2 specialist, available specialists per E2-X target, or six imports per E-T receiver target. This is the executed combined rule, not a substituted procedure or a lower bound on future gain.

Request continuation. Tables A8 and A9 give requests 1–12 and 13–24, respectively. Displayed request r is archive slot $r - 1$; splits do not reset the system. CARVE's storage after a request counts registered files only. E2 extension credentials add no model copy; each admitted specialist is 39,129,374 bytes. Service is timed separately from repair.

Table A8: Circuit revisits, requests 1 to 12. Quality uses 30 matched trials per method. Repair cost includes training and checks; storage follows each request.

Request	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Circuit	ibm07	ibm18	ibm15	ibm13	bigblue1	ibm13	ibm18	bigblue1	ibm07	ibm10	adaptec3	ibm13
Arrival	1	1	1	1	1	2	2	2	2	1	1	3
HPWL gain over Frozen (%)												
Overwrite	0.0	−0.6	0.8	4.5	4.8	−7.0	−11.5	3.4	−10.3	8.0	−37.0	4.9
RS	0.0	0.0	0.0	8.2	6.5	0.4	−20.7	9.3	−7.6	8.0	−15.4	3.9
Cache-only	0.0	0.0	0.0	0.0	0.0	17.0	−2.8	9.2	11.2	0.0	0.0	9.9
CARVE	0.0	0.0	0.0	0.0	0.0	20.7	17.6	9.5	16.3	0.0	0.0	16.0
Repair cost and CARVE storage												
Overwrite cost (min)	89.8	98.4	80.1	80.1	18.9	80.1	80.1	19.3	82.0	80.1	3.4	80.1
CARVE cost (min)	90.0	85.5	85.4	85.5	23.9	0.0	0.0	0.0	0.0	85.6	83.1	0.0
Storage (MiB)	37.3	74.6	112.0	149.3	186.6	186.6	186.6	186.6	186.6	223.9	223.9	223.9

Table A9: Circuit revisits, requests 13 to 24, continuing Table A8. Quality uses 30 matched trials per method. Repair cost includes training and checks; storage follows each request.

Request	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
Circuit	ibm18	ibm10	adaptec2	bigblue1	adaptec2	ibm15	adaptec3	ibm15	adaptec2	ibm10	adaptec3	ibm07
Arrival	3	2	1	3	2	2	2	3	3	3	3	3
HPWL gain over Frozen (%)												
Overwrite	−18.1	−3.6	−3.4	8.8	2.1	−7.9	−44.0	−13.3	−1.6	−24.9	−20.1	−17.0
RS	−9.8	31.2	−1.6	0.8	4.3	11.3	1.0	−6.5	−8.2	−3.1	8.4	−11.0
Cache-only	−7.1	8.1	0.0	14.8	5.4	31.2	3.2	24.0	10.3	15.5	7.7	11.5
CARVE	17.3	9.8	0.0	15.2	0.0	36.7	0.0	21.7	0.0	8.9	0.0	12.3
Repair cost and CARVE storage												
Overwrite cost (min)	80.1	81.8	3.9	19.0	0.0	10.5	0.0	0.0	0.0	96.1	0.0	89.7
CARVE cost (min)	0.0	0.0	82.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Storage (MiB)	223.9	223.9	223.9	223.9	223.9	223.9	223.9	223.9	223.9	223.9	223.9	223.9

Eight shared base files for the main extension sweep are charged once at slot 0; later sweeps add specialist files only. The allocation is retained in `timing_file_allocation.csv`. Timers exclude loading and setup as detailed in Appendix D.1; these are not end-to-end latency or resident-memory measurements.

A.4 Complete placement medians and training work

Table A10 consolidates the complete service-median and per-request training records. All 24 requests remain, including base first arrivals and harmful transfers. E2-X shares service seeds but changes training seeds and integration code; its column is a follow-up, not a sixth contemporaneous E2 arm. Values are rounded only for display; original CSV precision determines gains and ranks.

Table A10: Complete chip service medians and recorded training work. HPWL is in thousands and lower is better; training is in minutes and includes all recorded attempts.

			Median HPWL ($\times 10^3$)						Training (min)			
Req.	Circuit	Visit	B1	OV	RS	B3	CARVE	E2-X	OV	RS	B3	CARVE
1	ibm07	1	221.6	221.6	221.6	221.6	221.6	221.6	89.8	83.6	80.1	80.1
2	ibm18	1	566.4	570.0	566.4	566.4	566.4	566.4	98.4	80.1	94.8	80.1
3	ibm15	1	267.5	265.4	267.5	267.5	267.5	267.5	80.1	80.1	80.1	80.1
4	ibm13	1	265.4	253.5	243.6	265.4	265.4	265.4	80.1	152.3	117.7	80.1
5	bigblue1	1	329.9	314.2	308.5	329.9	329.9	329.9	18.9	18.5	19.0	18.9
6	ibm13	2	275.2	294.4	274.1	228.3	218.2	238.9	80.1	235.1	0.0	0.0
7	ibm18	2	561.9	626.8	678.4	577.6	463.2	487.4	80.1	80.1	0.0	0.0
8	bigblue1	2	322.8	311.8	292.7	293.2	292.2	316.5	19.3	0.0	0.0	0.0
9	ibm07	2	232.0	255.8	249.6	206.1	194.2	199.5	82.0	80.1	0.0	0.0
10	ibm10	1	1346.4	1239.3	1239.3	1346.4	1346.4	1346.4	80.1	80.1	80.1	80.1
11	adaptec3	1	3917.9	5366.3	4521.7	3917.9	3917.9	3917.9	3.4	80.1	80.2	80.2
12	ibm13	3	261.0	248.1	250.8	235.1	219.2	242.9	80.1	0.0	0.0	0.0
13	ibm18	3	551.7	651.8	605.5	590.9	456.1	481.9	80.1	187.9	0.0	0.0
14	ibm10	2	1293.8	1340.1	889.8	1189.4	1167.2	1254.6	81.8	0.0	0.0	0.0
15	adaptec2	1	10852.9	11217.7	11030.1	10852.9	10852.9	10852.9	3.9	80.1	80.1	80.1
16	bigblue1	3	344.5	314.0	341.8	293.7	292.1	323.4	19.0	18.9	0.0	0.0
17	adaptec2	2	11005.6	10770.2	10536.3	10409.9	11005.6	9229.9	0.0	80.1	0.0	0.0
18	ibm15	2	266.7	287.9	236.5	183.6	168.9	187.0	10.5	31.0	0.0	0.0
19	adaptec3	2	3930.7	5661.2	3892.4	3805.0	3930.7	3930.7	0.0	151.0	0.0	0.0
20	ibm15	3	254.9	288.8	271.5	193.6	199.7	192.8	0.0	0.0	0.0	0.0
21	adaptec2	3	10893.2	11066.3	11784.9	9769.4	10893.2	7916.0	0.0	104.3	0.0	0.0
22	ibm10	3	1334.7	1666.4	1375.9	1128.5	1215.9	1232.8	96.1	80.1	0.0	0.0
23	adaptec3	3	4347.5	5222.7	3984.2	4013.4	4347.5	4347.5	0.0	132.6	0.0	0.0
24	ibm07	3	228.0	266.9	253.1	201.9	199.9	200.6	89.7	80.1	0.0	0.0

Table A11 reports placement regressions and training outcomes. A regression is a request whose median HPWL exceeds Frozen by more than 5%. Each session permits up to three attempts and ends when an attempt produces a candidate. Rejection after evaluation does not permit another attempt.

Table A11: Placement regressions, training, and admission outcomes over 24 requests.

Method	$> 5\%$ worse (/24)	Sessions	Attempts	Candidates	Admissions
Frozen	0	0	0	0	—
Overwrite	11	19	32	16	Not tested
RS	7	20	33	18	13/18
Cache-only	1	8	11	8	Not tested
CARVE	0	8	8	8	6/8

Training totals include all recorded attempts, including those ending without a candidate. A circuit-registration issue in our integration affected some baseline attempts, so the comparison also reflects implementation effects.

Candidate outcomes also vary: E2 rejects adaptec2 at 7.04% gain ($p = .079$), while E2-X accepts another candidate at 21.22% ($p = .0073$). E2 accepts ibm18 at 19.78%, whereas three RS candidates for ibm18 are rejected. These comparisons vary in models and sometimes code, not seed alone. A rejection applies to the evaluated candidate, not to every model that could be trained for that family.

A.5 Compared deployment methods

Table A12 gives the update and model-selection rules. Repair retention uses Frozen, Overwrite, and CARVE. Caching and admission adds Cache-only, and chip placement also includes RS. Adapted CompoNet uses the same caching-and-admission arrival sequences with separate case blocks. The table describes the main and supplementary comparisons, not the earlier synthetic training protocol.

Table A12: Compared deployment methods. Each method serves the triggering request before training or admission. A repair can affect only later requests.

Method	Model update	Model used on later requests
Frozen	Never trains or changes the base.	Always uses the frozen base.
Overwrite	Fine-tunes the currently active model and replaces it without an admission test.	Uses the latest updated model for every family. Earlier models are not retrieved.
Cache-only	Trains a fresh copy of the base once for each family and saves the candidate without testing it.	Retrieves that family's saved candidate. Uses the base before a candidate exists.
RS	Trains a fresh copy of the base and tests it against the base. Acceptance replaces the active model; rejection keeps the previous one.	Uses one active model across families, without a family cache.
CompoNet (adapted)	Adds a PPO-trained module while freezing the base and earlier modules; no performance admission test.	Uses the family's module with its frozen prefix. Before repair, uses the frozen-base sampling wrapper.
CARVE	Keeps earlier specialists unchanged. Tests a new candidate against the base before saving it. Existing specialists may earn new families by evaluation.	Retrieves a specialist authorized for the requested family. Otherwise uses the base. A rejected family is not trained again within the stream.

A.6 Domain-specific contracts

Choice of admission margins. We selected the navigation and placement margins empirically from experiments conducted during development. Navigation requires at least 15 additional swept successes on 100 paired cases, equivalent to an absolute increase of 15 percentage points. Placement requires a relative reduction of at least 5% in median HPWL on 30 paired trials, together with the stated paired test and multiplicity rule. The placement effect margin is distinct from the test level $\alpha = 0.05$.

Evaluating a candidate before adoption has precedent in both empirical promotion rules and statistical policy improvement. AlphaGo Zero uses a win rate above 55% over 400 games against the current best player as its promotion criterion (Silver et al., 2017). High Confidence Policy Improvement allows the user to specify a performance requirement and statistical error tolerance separately (Thomas et al., 2015). The cited work supports the use of an explicit adoption criterion; CARVE's numerical margins remain choices from our development experiments. Our analysis holds the recorded margins fixed and does not establish an optimal choice. Section 2 and Appendix E.3 state the assumptions required for statistical guarantees and their relationship to the empirical rules.

Table A13 gives the task definitions and repair rules used in each domain. In the navigation alarm, k counts swept successes among 100 episodes. The synthetic chip alarm belongs to the earlier study, while E2 uses the model or family identity. The chip selector row describes main E2. E2-X and E-T instead use the recorded earliest passing specialist as described in Appendices B.2 and B.3. The budget $r = 3$ permits at most three attempts in one session, including the first attempt. Exact admission and extension procedures remain in Appendix A.3.

Table A13: Domain-specific task definitions and repair rules.

Component	Crowd navigation	Chip macro placement
Family key	Environment geometry and crowd-behavior model	Circuit identity with macro and netlist specification
Governed score	Swept-contact success; higher is better	Half-perimeter wirelength (HPWL); lower is better
Alarm	Statistical: 100-episode block; Clopper–Pearson rule equivalent to $k \leq 88$	Synthetic families: random-anchor normalized-improvement statistic below 0.7585. Real circuits: no statistical alarm fired; identity-triggered (no credential $\Rightarrow$ alarm), see Appendix D.4
Candidate	SARL specialist trained from the frozen base on the current family	ChiPFormer specialist fine-tuned from the frozen base on the current circuit; $4{,}800$ s budget, at most $r{=}3$ training attempts
Admission	Paired 100-episode block; at least $15/100$ swept successes over the base	Paired 30-episode block; median HPWL improved by at least 5% and one-sided Wilcoxon signed-rank $p < 0.05$
Extension exam	Before training, on the alarmed family's paired block, same 15 rule; a pass routes the family	Before training (exam-routing arm) or after admission with credentials logged (main arm); Holm over the simultaneous decisions
Selector σ	Own-family specialist first, then earliest credential	Own-family specialist first (the only case exercised)
Fallback	Byte-identical frozen SARL base	Byte-identical frozen ChiPFormer base

A.7 EXPERIMENTAL TERMS AND REPORTED QUANTITIES

Table A14 brings together definitions used in the text, tables, and logs. The definitions do not add new metrics or alter the recorded measurements.

Table A14: Terms used in the experiments. Requests, training attempts, and evaluation cases are different counting units.

Term	Meaning in the recorded experiments
Family	A navigation geometry and pedestrian model, or a chip circuit with its macro and netlist specification. Circle crossing with ORCA+ and circle crossing with SFM are different families.
Scene setting and case	A setting fixes parameters within a navigation family, such as scene size and pedestrian count. A case also has an evaluation index. The index alone does not identify a case.
Request	In the main comparisons, one family arrival contains 100 navigation cases or 30 chip placement trials per method.
Stream and repair epoch	A stream is the ordered request list. Each main stream starts a separate repair epoch, with the same base and an empty repair state. A later stream does not continue an earlier one.
Arrival seed	The random seed that generates the request order and scene settings. It is distinct from the seed used to train a model. The displayed caching and admission stream uses fixed per-family request counts.
Base and specialist	The base is the original frozen policy. A specialist is a separate trained policy. An CARVE specialist remains unchanged after admission.
Admission and credential	Admission evaluates a candidate against the base under the recorded rule. A credential records permission to use a specialist for a specific family and metric. Passing on one family does not authorize another.
Extension and reuse	An extension evaluates a saved specialist on another family without changing its weights. A pass adds permission for later requests. Reuse means serving a request with a saved model.
Rejection and fallback	Rejection excludes a candidate from CARVE service. CARVE serves the base for that family and records that it will not train again within the epoch. RS instead retains its previous active model.
Training session and attempt	A session is one repair training job. In main chip E2, it permits up to three attempts and ends when an attempt produces a candidate. An attempt that produces no candidate is distinct from a candidate rejected after evaluation. A session with no candidate after its allowed attempts is exhausted.
Swept success	The robot reaches its goal within 90 seconds without swept-volume contact. Goal reaching without the contact condition is a different score.
HPWL and gain	HPWL is estimated wirelength, with lower values preferred. Request gain is $100(1 - m/m_0)$, where m and m_0 are the method and Frozen medians on matched trials. Positive gain means shorter estimated wiring.
Mean rank and worse requests	Rank is one plus the number of methods with strictly lower median HPWL on the same request. Ties share a rank. Mean rank averages the request ranks. The $> 5\%$ worse count uses requests with median HPWL above $1.05m_0$.
Mean $\pm$ SD and pp	The main stream SD is variation across matched requests, not independent training runs. Navigation differences in pp are percentage-point differences in success rate. Receiver intervals use a separate definition in Appendix D.2.
Repair and service time	Repair time sums training and recorded admission and extension checks, including failures. Ordinary service is separate. Chip trial timers exclude model loading and do not measure full request latency.
Checkpoint storage	Logical sizes of final registered specialist files. It is not RAM or GPU memory. Duplicate extension copies are included in the cache comparison, with the full counting rules in Appendix D.1.

A.8 REPAIR PROTOCOL DETAILS

Scope and repair epochs. A repair epoch is one deployment run with fixed versions of the base, family resolver, metric, and repair contract. Each experimental request stream starts a separate epoch. Runs with the same versions still maintain separate credentials, stopped-family records, and ledgers. Replacing the base, changing the family definitions, or changing a decision rule starts a new epoch. Earlier specialists and their evidence remain available. An earlier credential applies to its original base and contract. A saved specialist can obtain a credential under a new base by passing another evaluation, without changing its weights.

Required interfaces. The operator supplies an executable frozen base M_0 and a deterministic resolver $\rho(x) \in \mathcal{F} \cup \{\perp\}$. The resolver assigns request descriptor x to a declared family or reports that it is outside scope. The operator also supplies a score evaluator, a candidate trainer, and storage for model artifacts and evidence. The score $s_f(M, z)$ evaluates model M on instance z from family f, with larger values preferred. A quantity with smaller preferred values, such as HPWL, uses a fixed orientation or an equivalent predicate for that metric. The trainer uses the base and current-family data. No source training set, source replay buffer, or samples from previous families are required.

The resolver uses observable task keys. Navigation keys specify geometry and the pedestrian model. Placement keys specify the circuit and its macro and netlist description. The experiments do not learn a resolver for ambiguous or continuously changing conditions. An unresolved request uses the base without approximate matching to a saved specialist.

State and model selection. At time t, the deployment maintains

$$\begin{aligned} &\mathcal{M}_t = \{M_0, M_1, \ldots, M_{J_t}\}, \qquad \Gamma_t \subseteq \{1, \ldots, J_t\} \times \mathcal{F}, \\ &\mathcal{T}_t \subseteq \mathcal{F}, \qquad \mathcal{X}_t \subseteq \mathcal{F}, \qquad \mathcal{L}_t = \text{append-only audit ledger.} \end{aligned} \tag{4}$$

The model library $\mathcal{M}_t$ contains immutable artifacts. A credential $(j, f) \in \Gamma_t$ authorizes specialist M_j for family f. One specialist can serve several families, and several specialists can qualify for the same family. The contract fixes a deterministic selector $\sigma(f, \Gamma_t)$ for the latter case. Navigation and the main placement study prefer a specialist trained on the requested family, then the earliest-admitted eligible credential. The placement reuse follow-up selects the earliest-admitted specialist among those that pass. The receiving operator selects the passing import admitted earliest at the source, as specified in Appendices B.2 and B.3.

The tombstone set $\mathcal{T}_t$ records families whose new candidate received a complete, valid rejection. The exhausted set $\mathcal{X}_t$ records families whose training session ended without a candidate. Exhaustion is a training failure rather than a performance verdict. Both sets prevent another training session for that family in the epoch. A failed extension exam alone does not create a tombstone, since a new candidate may still be trained. An incomplete gate record cannot add an admitted artifact, credential, or performance tombstone.

Serving and monitoring requests. The system serves each request before any repair triggered by that request. For a credentialed family, the selector retrieves an authorized unchanged specialist without training or another admission test. For an unresolved, tombstoned, or exhausted family, the system serves the exact frozen base. An eligible family without a credential also uses the base for its current request. A repair can therefore improve only later requests, as in the other repairing methods in Section 3.

After base service for an eligible family, the adapter computes

$$a_t = \mathsf{Alarm}_{f_t}(M_0; E_t, \mathcal{C}), \tag{5}$$

where E_t is the monitoring block and $\mathcal{C}$ is the frozen contract. The contract states the score, sampling unit, decision rule, and validity checks. It also states the uncertainty calculation when one is used. A complete, valid alarm opens a repair event. A no-alarm result leaves the family eligible for monitoring at its next arrival. Incomplete monitoring evidence commits no state change and leaves the base route in place. Navigation uses a statistical alarm. The placement experiments on benchmark circuits instead use the recorded model or family identity rules, as documented in Appendix D.4.

If the contract monitors a credentialed family, a later alarm is logged without another training session in that epoch. A credential is removed only through an explicit logged revocation or an epoch change.

Extension exams and their order. For family f, the contract fixes an ordered set $\mathcal{H}_f$ of saved specialists eligible for an extension exam. Each specialist is evaluated against M_0 using the same paired admission rule as a new candidate. A pass adds (j, f) to Γ_t without changing any weights. The contract fixes the comparison order and the correction for multiple comparisons before evaluation. Appendix A.3 records the executed placement rule, including its effect requirement and successive Holm thresholds. The comparison set depends on the study.

Navigation and the placement reuse follow-up examine saved models before training. A passing specialist can then serve later requests without new training. If none passes, an eligible family receives a training session. The main placement study instead gives each new circuit its own training session and examines an admitted specialist on all eight circuits afterward. Those additional credentials are recorded as transfer evidence and do not avoid training or change routing in that study. Appendix B.2 keeps the two schedules separate.

Models received from another operator. An imported specialist begins without permission to serve the receiver's families. The receiver compares it with its own frozen base on a local paired evaluation block. A pass creates a local credential, while a failed exam leaves the base route available. A source credential does not replace the receiver's evaluation. Requests assigned a different family require another local exam even when they resemble a source task. Reuse therefore requires local evaluation data but no source circuits, source training histories, or weight updates. The receiving-operator study evaluates its selected model on a separate service block (Appendix B.3).

Candidate generation and evaluation data. Each candidate starts from a byte-identical copy of the frozen base:

$$C_f = \mathsf{Train}_f(M_0, D_f^{\mathrm{tr}}; \xi_f). \tag{6}$$

The data D_f^{tr} come from the current family, and ξ_f specifies training settings, resources, and random seeds. The trainer may use full fine-tuning, parameter-efficient adaptation, online reinforcement learning, or another local update. It must produce a serializable policy and support evaluation under the contract. Candidates cannot serve while training or evaluation is incomplete. Training leaves the base and every saved specialist unchanged, so retaining earlier models requires no previous-family examples.

The intended contract separates admission data D_f^{dev} from candidate training. Its statistical guarantees require the validation assumptions in Section 2. Appendix C.2 describes case allocation and the available evidence for the recorded navigation runs.

Paired admission and statistical interpretation. For evaluation cases $z_1, \ldots, z_n$, candidate and base produce paired differences

$$d_i = s_f(C_f, z_i) - s_f(M_0, z_i). \tag{7}$$

The contract fixes an effect estimator $\widehat{\Delta}_f$, a minimum margin m_f, an uncertainty procedure when used, and an acceptance region. A contract with both an effect requirement and a statistical test takes the form

$$\mathsf{Admit}_f(C_f, M_0; D_f^{\mathrm{dev}}) = \mathbf{1}\{\widehat{\Delta}_f \geq m_f \text{ and } p_f \leq \alpha_f\}. \tag{8}$$

Here p_f is the test's p-value and α_f is its declared threshold. A contract can instead specify a paired margin-only rule or an equivalent confidence-bound rule. Score orientation, comparison unit, effect margin, error criterion, evaluation block, and multiplicity policy are fixed before examining the candidate.

The effect estimator can summarize paired differences or relative change in medians. Placement measures one minus the ratio of the candidate and base HPWL medians and applies a one-sided Wilcoxon signed-rank test to paired outcomes. Navigation applies its paired improvement margin without that test. Appendix A.6 gives the domain thresholds, and Appendix A.3 gives the executed placement rule.

A passing evaluation authorizes only the tested family and metric. Under a valid level-α_f test and its sampling assumptions, requiring rejection of the declared null controls erroneous rejection of that null at the corresponding level, subject to the declared multiplicity rule. An empirical effect margin alone supplies no such error control. Neither rule establishes universal policy safety, optimality, or a minimum gain on every future instance. In particular, the empirical margin is not automatically a lower bound on expected improvement. Section 2 and Appendix E.3 separately state the conditions for the bounded-loss guarantees and their relation to the experiments.

Training attempts and complete decisions. Each family receives at most one bounded training session in an epoch. A session allows up to r attempts under a predeclared retry schedule. Only an attempt that ends without a candidate may be retried with the next seed. Producing a candidate ends the session, and the candidate then receives one gate decision. A performance rejection cannot start another attempt. Using all attempts without producing a candidate exhausts the session.

Acceptance serializes and hashes the candidate, marks it immutable, and records its credential, contract identity, and complete evaluation evidence in one atomic commit. Only then can the candidate serve requests. The admitted specialist may obtain further credentials through extension exams without changing its parameters. A complete failed gate adds the family to $\mathcal{T}_t$ and leaves M_0 serving. The rejected artifact may remain in an archive for analysis but cannot have an active credential.

An exhausted session adds the family to $\mathcal{X}_t$ and records each attempt's failure evidence. No candidate was evaluated, and the family uses M_0 for the rest of the epoch. Incomplete evaluation evidence cannot authorize a model or record a performance rejection. The navigation contract used $r = 1$ and would halt the epoch after a training failure; none occurred. Placement permits up to three attempts per session and distinguishes sessions that produce no candidate from candidates rejected after evaluation.

Automation and provenance. The operator defines scope, family resolution, metrics, decision rules, the trainer, resource budgets, retry limits, and epoch changes. Once the contract is fixed, CARVE executes the request and repair procedure. It resolves and serves the request, then monitors the outcome under the contract. When repair is triggered, it runs the specified extension or training sequence and evaluates the candidate. It records acceptance, rejection, or exhaustion before changing subsequent routing. The validator and admission procedure control entry into service.

The ledger records request and family identifiers, base and candidate hashes, contract and code versions, training and evaluation block identifiers, and seeds. It also records per-instance and aggregate scores, uncertainty calculations, decisions, selected routes, wall-clock costs, and failure evidence. Model identities support operational rollback, and evaluation records support recomputation of the reported decisions.

A.9 Construction properties and resource accounting

Preserving models and controlling service. Within an epoch, training a candidate cannot modify M_0 or a previously admitted specialist. When inference settings are also fixed, later training therefore leaves the saved policy unchanged. Deleting an artifact is a storage failure. Compressing or replacing it creates a different artifact that requires a new identity and authorization. Preserving parameters does not keep realized performance constant across changing evaluation cases or task distributions.

A candidate cannot serve before its credential is committed. An out-of-scope, tombstoned, or exhausted family uses the exact frozen base. The construction guarantees concern preservation and selection. They do not establish adequate base performance or improvement by a specialist on every request. Section 2 gives conditions for preserving expected performance relative to the base. Later alarms on a credentialed family are logged without further training within the same epoch.

Without an explicit logged revocation or an epoch change, Γ_t only gains authorized pairs. An extension can therefore increase the recorded scope of reuse without adding model weights. The original artifact and its evaluation record remain available for earlier authorized uses.

Training work and storage. Let K count distinct alarmed families in a finite epoch, let J count accepted newly trained specialists, and let r be the maximum number of training attempts per session. The one-session rule gives

$$N_{\text{sessions}} \le K, \qquad N_{\text{runs}} \le rK, \qquad J \le K. \tag{9}$$

A method that retrains after each alarm can start several sessions for the same family. Equation 9 bounds sessions and training attempts rather than extension exams or their cost. Every permitted attempt remains subject to its declared resource budget.

For a local library containing the base and J accepted repairs, with one stored copy of each model, logical serving storage is

$$B_{\text{serve}} = B(M_0) + \sum_{j=1}^{J} B(M_j) + O(|\Gamma|), \tag{10}$$

where $B(M)$ is model size and $|\Gamma|$ counts credentials. Imports and duplicate physical files require additional storage. Appendix D.1 defines the measured file totals. Full checkpoints, adapters, compression, and loading policies have different costs, and transformed artifacts require new identities and authorization. The bound depends on the declared scope. The model library can grow as the number of families grows.

B REUSE AND TRANSFER RECORDS

The records below distinguish checks that authorize reuse from later service with the same saved model. Request counts do not represent independent training runs.

B.1 NAVIGATION EXTENSION DECISIONS

The four displayed successful extensions in Table B1 add permissions without updating weights. They include ORCA+/SFM changes and square crossing. The subsequent-service summary uses the same retrospective subset outside the permitted training index range as Appendix C.2. The three detailed streams contain 35 own-family and 21 extension requests with unchanged model identities, not new training repetitions.

Table B1: Successful navigation extension exams. Time covers these successful events only; full repair costs also include failed checks.

Stream	Slot	Target family	Specialist	Base	Exam time (s)
Fixed arrivals	5	F-CS	96/100	79/100	88.5
Sampled arrivals	3	F-CS	95/100	26/100	302.8
Caching and admission	3	F-CO	96/100	5/100	351.9
Caching and admission	9	F-SQ	56/100	30/100	223.6

Table B2: Later service using retained specialists, relative to matched Frozen (pp). Ranges show observed requests, not confidence intervals.

	Own training family			Extension family		
Stream	n	Mean	Range	n	Mean	Range
Fixed arrivals	17	38.18	[−6, 88]	6	41.50	[20, 67]
Sampled arrivals	14	58.29	[25, 86]	6	47.17	[18, 66]
Caching and admission	4	18.25	[−4, 36]	9	27.11	[8, 65]

All 56 requests are from that retrospective subset. Earlier development exposure is not excluded.

Earlier navigation checks. Development evaluations also compared one specialist on three case banks and on ORCA+ and SFM pedestrian grids. They used fixed models and did not authorize service or use the reserved final SFM exam. Their complete records remain in `analysis/appendix_original_upload.tex`.

B.2 PLACEMENT REUSE BEFORE TRAINING

The reuse study follows the main circuit order and service seeds, but checks saved specialists before training. The base serves each first arrival. A passing check permits the earliest admitted qualifying specialist to serve later arrivals unchanged. Five circuits reuse the `ibm07` specialist, while `ibm07`, `adaptec3`, and `adaptec2` receive training sessions. Training seeds and circuit integration code differ between the executions, so their comparison does not isolate extension timing.

Table B3: New-circuit checks and later outcomes. Jobs count training sessions; service gains average the two revisits relative to Frozen.

	Check saved model		Train each circuit		Check saved models first		
	Exam gain			Service gain			Service gain
Circuit	(%)	Result	Jobs	(%)	Jobs	Model	(%)
ibm07	—	None	1	14.29	1	ibm07	13.01
ibm18	14.1	Pass	1	17.44	0	ibm07	12.96
ibm15	26.9	Pass	1	29.17	0	ibm07	27.14
ibm13	10.0	Pass	1	18.36	0	ibm07	10.08
bigblue1	10.1	Pass	1	12.35	0	ibm07	4.05
ibm10	6.8	Pass	1	9.34	0	ibm07	5.34
adaptec3	−3.1	Fail	1	0.00	1	Base	0.00
adaptec2	−2.6	Fail	1	0.00	1	adaptec2	21.73

Figure 4 includes all 24 requests, including first arrivals and base fallbacks. For request r, the gain is $g_r = 100(1 - m_r/b_r)$, where m_r and b_r are the method and Frozen service medians. Mean gains across all 24 requests are 8.41% for training each circuit and 7.86% for checking saved models first.

Recorded training times are 9.657 and 4.005 hours, respectively. Including checks gives 10.362 and 4.247 hours under Appendix D.1's timing definitions. The `adaptec2` gains at requests 17 and 21 come from new training.

On the ten revisits to the five reused circuits, mean gains are 17.33% for own-circuit training and 11.91% for reuse. Reuse gives lower gains on nine of those ten requests. Appendix Table D6 retains every paired revisit. The results therefore support reduced training work with a placement-quality trade-off.

Extension coverage and recorded predictions. The main study's six saved specialists pass 23 model–family checks across six targets, with no pass on `adaptec2` or `adaptec3`. Those checks occurred after training and did not avoid any training session. The reuse study did not meet its recorded prediction of mean rank no worse than Cache-only. Its six-method mean rank was 2.375, compared with 2.208 for Cache-only. The complete coverage and prediction records remain in `analysis/appendix_original_upload.tex`.

B.3 Receiving-operator selection and service

The receiver imports six admitted source checkpoints and tests them against its base on seven new IBM targets and an `adaptec1` control. Each comparison uses 30 paired exam trials. The earliest source-admitted passing model is selected, and its service uses a separate block of 30 seeds. The receiver trains no weights and uses no source circuit files or training trajectories. The experiment separates operator task sets without enforcing privacy isolation. A prior census evaluated `ibm06` through `ibm18`; five IBM circuits entered the source study, and `ibm06` was an integration probe, leaving seven receiver targets.

Training the six imported source specialists cost 419.17 minutes in total, paid once per specialist. Receiver checks cost 20.71 minutes across all eight targets, with no receiver training. Each target requires one base and six import evaluations of 30 trials each. No matched receiver training arm was run, so the source time does not estimate receiver training time or time saved. Source training includes work inside the training process, while receiver check timers exclude model loading and setup (Appendix D.1). The reported 5.76% mean gain averages all eight target gains, including both base fallbacks.

Table B4: All 48 local exam gains (%). Bold entries mark passing under the recorded combined effect-and-Holm rule; signs alone do not determine admission.

Target	ibm07	ibm18	ibm15	ibm13	bigblue1	ibm10
ibm08	0.51	−2.89	−2.62	−1.18	−15.62	−1.63
ibm09	**6.45**	**6.09**	−5.64	**15.02**	−1.38	2.38
ibm11	**9.31**	**7.35**	−1.00	**14.06**	−3.91	2.85
ibm12	3.82	3.47	−1.09	**8.71**	−5.81	4.15
ibm14	3.73	**5.54**	−1.63	1.74	−2.76	−0.09
ibm16	**14.14**	**8.48**	1.15	**7.75**	−1.32	**6.53**
ibm17	**10.87**	**15.07**	−5.84	8.22	6.80	5.98
adaptec1	−13.05	−5.70	−3.03	−8.35	−8.60	−6.37

Table B4 contains exam effects and decisions. Fourteen passing model–target pairs cover six targets. Table B5 reports the selected routes and their later service, including both fallbacks.

Table B5: Local selection and later service. All base fallbacks and both intervals containing zero remain. Intervals condition on the selected model and target.

Target	Passing imports	Selected	Gain (%)	95% interval	Lower/equal/higher
ibm08	—	base	0.00	—	—
ibm09	ibm13, ibm07, ibm18	ibm07	3.61	[−2.00, 7.09]	18/0/12
ibm11	ibm13, ibm07, ibm18	ibm07	10.21	[4.28, 14.91]	22/0/8
ibm12	ibm13	ibm13	9.95	[2.02, 15.51]	21/0/9
ibm14	ibm18	ibm18	8.92	[2.46, 14.21]	23/0/7
ibm16	ibm07, ibm13, ibm18, ibm10	ibm07	8.06	[3.66, 11.98]	21/0/9
ibm17	ibm18, ibm07	ibm07	5.33	[−3.36, 11.27]	19/0/11
adaptec1	—	base	0.00	—	—

Checks and later service. Figure B1 compares the exam with later service for all five reused circuits and all six selected receiver imports. Some later gains fall below the 5% required in the

check, so passing does not ensure the same gain on later cases. All selected receiver imports have positive service point estimates, but the intervals for `ibm09` and `ibm17` include zero.

On `ibm09`, the selected `ibm07` model improves HPWL by 6.45% in the exam and 3.61% in service. The unselected `ibm13` import has a 15.02% exam gain but no corresponding service evaluation, so the two blocks cannot estimate selector regret. Appendix D.2 defines the conditional paired-bootstrap intervals.

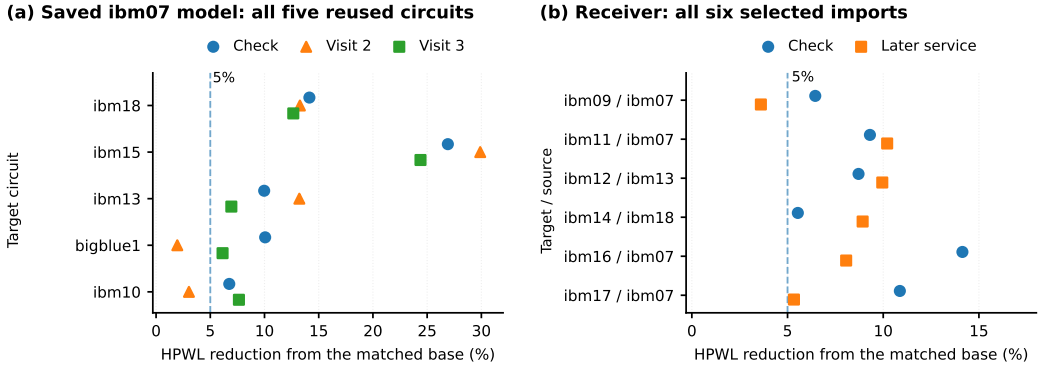


Figure B1: Checking and later using the same saved model on the same target. The left panel includes every circuit served by reuse of `ibm07`; each later marker represents one 30-trial revisit. The right panel includes every selected import at the receiver, evaluated on a separate 30-trial block. The dashed 5% line applies to the check, not to later service.

C Ablations and Component Records

The following records examine admission decisions and model selection. Comparisons using the same candidate separate deployment decisions from training variation. Full traces retain both beneficial and harmful outcomes.

C.1 Navigation Decisions and Subsequent Service

A matched candidate, with deployment or fallback. Saving a candidate does not establish that it should replace the base. We examine the hallway candidate in the displayed caching and admission stream. Its independently trained Cache-only and CARVE model states match. Cache-only serves the candidate, whereas CARVE rejects it and serves the base. We compare all five later hallway requests, each with 100 matched cases. Admission uses a different 100-case block and requires at least 15 more successes than the base. The candidate gains 10 and therefore fails this requirement.

Table C1: Serving the matched hallway candidate or retaining the base in the displayed stream. Later scores pool five requests with 100 cases each. The final column is CARVE minus Cache-only in percentage points.

Task	Candidate gain in check (pp)	Cache-only: use candidate Later successes /500	CARVE: keep base Later successes /500	CARVE minus Cache-only (pp)
Bottleneck hallway	10	168	195	5.4

Table C1 compares serving the same candidate with retaining the base. Rejection avoids a loss over these later hallway requests. The result concerns this candidate and arrival history; it does not establish that rejection always improves service. The later evaluation below uses the retrospective subset outside the permitted training index range (Appendix C.2), so its request counts and mean gains differ from the five-request totals above.

We compare the seven admission decisions in the three detailed streams with later evaluation of the same candidates. Accepted states match CARVE service. The rejected caching-and-admission hallway state matches the independently trained Cache-only model. The stream with sampled arrivals has no Cache-only arm, so its rejected hallway candidate uses the executed recheck outside the permitted training index range. A fallback score is never substituted for the rejected candidate's performance. Table C2 reports all seven decisions and their later evaluations.

Table C2: Seven navigation gates and later own-family evaluation on the retrospective subset outside the permitted training index range. Gains are successes per 100; blocks do not represent independent training runs.

Stream	Family	Gate slot	Decision	Gate gain	Blocks	Evaluated via	Later mean gain
Fixed arrivals	F-CO	1	Accept	59	12	CARVE	52.92
Fixed arrivals	F-HO	3	Accept	15	5	CARVE	2.80
Sampled arrivals	F-CO	1	Accept	89	14	CARVE	58.29
Sampled arrivals	F-HO	6	Reject	10	1	Sampled-arrival recheck	6.00
Caching and admission	F-CS	1	Accept	64	2	CARVE	36.00
Caching and admission	F-HO	7	Reject	10	4	B3	−5.00
Caching and admission	F-HS	8	Accept	22	2	CARVE	0.50

In caching and admission, the rejected hallway model changes success by $-49, +15, +6, +8, -7$ over its five later requests. The negative average coexists with gains on three requests. On request 10, Frozen and CARVE score 51/100, Cache-only scores 2/100, and Overwrite scores 28/100 on the same recorded cases. The sampled-arrival recheck still shows a useful candidate rejected by the fixed margin (Appendix C.3).

C.2 Navigation evaluation scope

The three detailed streams contain 300 service files and 30,000 episode executions. Admission and service use different index blocks. The admission block and some service indices lie within the permitted training range. A case is identified by its family, scene settings, and index, so overlapping indices alone do not establish that the same cases were used in training.

The subset outside the permitted training index range covers 76 shared request positions and 24,900 method–episode executions. The other 5,100 executions fall within that range. The archive retains all 420 files, including the additional arrival. Counts refer to executions, not unique environments.

Table C3 applies the same retrospective filter to every method while retaining the executed models and routes. Earlier requests may have affected later decisions, and prior development exposure is not excluded. This analysis describes existing service outcomes.

Table C3: Service outcomes on the retrospective subset outside the permitted training index range. Original training and routing remain unchanged.

Stream	Requests	Episodes/arm	Frozen	Overwrite	B3	CARVE
Fixed arrivals	26	2,600	.366	.631	—	.711
Sampled arrivals	29	2,900	.333	.699	—	.712
Caching and admission	21	2,100	.425	.577	.554	.576

The displayed subset contains 56 credential hits, with 35 on the five accepted candidates' training families and 21 after extension. Every accepted candidate has later own-family service with a matching model state. The caching and admission subset retains the avoided loss on request 10, but CARVE's full-stream advantage over Overwrite is absent. Its admitted constricted-hallway model also scores 53 against the base's 57 on request 22. The subset establishes neither equivalence nor absence of harm.

C.3 Rechecking the hallway model rejected under sampled arrivals

The fixed hallway model rejected under sampled arrivals is re-evaluated against the base on two paired blocks. One block lies within the permitted training index range; the other lies outside it, although its base results already existed. Both yield modest positive swept gains (Table C4). Using a base score from another block would overstate the effect. These are two evaluations, not new training runs or permanent bounds on improvement.

Table C4: Matched rechecks of the hallway model rejected under sampled arrivals. Gained/lost cases preserve the pairing; scores use saved outcomes.

Case block	Base	Candidate	Gain	Gained cases	Lost cases
Within permitted training range	32/100	39/100	7	13	6
Outside permitted training range	28/100	34/100	6	15	9

C.4 PLACEMENT ADMISSION AND MODEL SELECTION

Admission on fixed candidates. We compare the recorded rule with a rule requiring only 5% observed median-HPWL improvement on the same eight trained candidates. The recorded rule also requires a passing paired statistical check. Removing that check admits one additional candidate, `adaptec2`, whose 7.04% observed gain has $p = 0.079$. The comparison identifies a change in admission decisions. Neither rejected candidate has a later candidate-service evaluation, so the comparison cannot establish the service effect of admitting it. The margin remains fixed throughout the audit, and a passing check does not establish a population lower bound of 5% gain. Appendix A.3 defines the recorded rule.

Table C5: All placement candidates under an effect-only rule and the recorded rule. Checks use 30 paired trials. Later gains use the admitted model; rejected candidates were not evaluated in later service.

Slot	Circuit	Exam gain (%)	One-sided p	Paired wins	Effect only	Recorded rule	Later gain (%) visit 2 / visit 3
0	ibm07	11.53	9.96×10^{-7}	27/30	Accept	Accept	16.27/12.31
1	ibm18	19.78	1.77×10^{-8}	28/30	Accept	Accept	17.57/17.32
2	ibm15	30.56	1.95×10^{-5}	25/30	Accept	Accept	36.69/21.65
3	ibm13	19.24	1.86×10^{-9}	29/30	Accept	Accept	20.71/16.01
4	bigblue1	10.38	9.31×10^{-10}	30/30	Accept	Accept	9.48/15.22
9	ibm10	12.83	1.18×10^{-5}	24/30	Accept	Accept	9.78/8.90
10	adaptec3	4.60	0.657	12/30	Reject	Reject	Not evaluated
14	adaptec2	7.04	0.079	17/30	Accept	Reject	Not evaluated

Model selection on arriving circuits. Table 6 shows four consecutive requests for which RS uses its `bigblue1` specialist and CARVE retrieves each circuit's earlier validated repair.

The complete RS trace in Table C6 includes every request. All seven requests with HPWL more than 5% worse than Frozen use a model on a different circuit from its admission circuit. Ten other such uses improve on Frozen, so the record does not show that all reuse across circuits is harmful.

Table C6: Complete RS trace. Positive HPWL change is worse than the matched base. Same/Other refers to the model's earlier admission family.

Slot	Requested	Model origin	Admit slot	Scope	HPWL increase	Harm $> 5\%$
0	ibm07	base	—	Base	0.00%	No
1	ibm18	base	—	Base	0.00%	No
2	ibm15	base	—	Base	0.00%	No
3	ibm13	ibm15	2	Other	−8.20%	No
4	bigblue1	ibm13	3	Other	−6.48%	No
5	ibm13	bigblue1	4	Other	−0.39%	No
6	ibm18	bigblue1	4	Other	20.73%	Yes
7	bigblue1	bigblue1	4	Same	−9.33%	No
8	ibm07	bigblue1	4	Other	7.61%	Yes
9	ibm10	ibm07	8	Other	−7.95%	No
10	adaptec3	ibm10	9	Other	15.41%	Yes
11	ibm13	ibm10	9	Other	−3.91%	No
12	ibm18	ibm10	9	Other	9.75%	Yes
13	ibm10	ibm10	9	Same	−31.23%	No
14	adaptec2	ibm10	9	Other	1.63%	No
15	bigblue1	adaptec2	14	Other	−0.79%	No
16	adaptec2	bigblue1	15	Other	−4.26%	No
17	ibm15	adaptec2	16	Other	−11.34%	No
18	adaptec3	adaptec2	16	Other	−0.97%	No
19	ibm15	adaptec3	18	Other	6.49%	Yes
20	adaptec2	adaptec3	18	Other	8.19%	Yes
21	ibm10	adaptec2	20	Other	3.08%	No
22	adaptec3	ibm10	21	Other	−8.36%	No
23	ibm07	adaptec3	22	Other	11.00%	Yes

Trace verification and scope. Every non-base route has a 16-hex hash prefix matching the preceding accepted candidate in the logs. The trace contains three base routes, two routes on the admission circuit, and 19 routes on other circuits. Adding a scope check could change later training and model selection, and that alternative was not run. The trace therefore does not isolate the causal benefit of the check. Hash agreement verifies recorded identities without supplying missing checkpoint files.

C.5 DEVELOPMENT RECORDS

Earlier recipe, scene-screening, implementation, and synthetic-placement studies remain together in `analysis/appendix_original_upload.tex`. The archived block retains successful, failed, incomplete, and unfavorable results under their original metrics and protocols. They are development records and do not add independent runs to the benchmark comparisons.

Navigation recipe screening selected the local trainer and used separate confirmation seeds. Its historical goal-reaching criterion differs from the main swept-success metric. A three-seed base-penalty study reduced hallway performance on every seed. Routing tests, unsuccessful action diagnostics, and exploratory scene grids are retained in the same archive.

Earlier synthetic placement studies used different service order, baselines, and score definitions. Their transfer checks include both useful and harmful outcomes. A later overwrite study did not confirm the predicted home-task degradation. Earlier circuit-specific qualification runs used different candidates from the benchmark studies, so their outcomes are kept separate.

D MEASUREMENT DEFINITIONS AND STATISTICAL ANALYSES

This section specifies what is counted and what variation means. Calculations use retained execution records, without new policy training or simulation. Training runs, requests, evaluation trials, and files are different units and are not added into one sample count.

D.1 REPAIR-TIME AND CHECKPOINT-STORAGE ACCOUNTING

Timing boundaries. Navigation repair time sums training, admission, and extension process times, including failed checks; ordinary service and driver work are excluded. The detailed tables contain ten original method–stream totals; the unchanged archive also preserves the four totals from the additional arrival. Placement training time includes all recorded attempts and evaluation inside the training process. Separate evaluation-trial time starts after model loading and trainer construction: seeding, placement, and the recorded `wall_s` are timed, with values rounded to .01 seconds. These timers omit Python startup, deserialization, device setup, overwritten evaluation outputs, and interrupted work absent from retained files. No complete cold/hot loading, peak CPU/GPU memory, throughput, or end-to-end latency profile was collected.

Table D1: Exact-phase CARVE navigation totals, rounded after summation. Checks include failed extensions; repair excludes ordinary service.

Stream	Trainings	Train (h)	Checks (h)	Repair (h)	Checks/repair
Fixed arrivals	2	1.394	0.340	1.734	19.6%
Sampled arrivals	2	1.301	0.294	1.596	18.4%
Caching and admission	3	2.235	0.729	2.964	24.6%

Table D2: Placement training and separately timed gate/extension work. Totals include all recorded training attempts; shared base evaluation files are counted once per study.

Study / arm	Training (h)	Gate trials (min)	Extension trials (min)
E2 B1	0.000	0.00	0.00
E2 OV	19.556	0.00	0.00
E2 RS	31.933	24.58	0.00
E2 B3	10.534	0.00	0.00
E2 CARVE	9.657	9.44	32.81
E2-X	4.005	6.06	8.46
E-T receiver	0.000	0.00	20.71

Shared files and internal evaluation. E2's 48 extension comparisons use eight shared base and 48 specialist files, not 96 unique runs. E2-X uses seven base/specialist pairs; E-T uses eight base plus 48 import files. Base files are counted once within each study, never merged across studies. E-T service consists of 240 base and 180 selected-import trials, not 420 end-user requests. E2's eight random-anchor files add 240 diagnostic trials totaling 19.69 seconds and one metadata row per file; those are not assigned to an individual method. Fifty successful E2 training processes include 100

internal evaluation placements, and E2-X adds six. Their time is already inside training and is not added again. The sealed lost-interruption field is zero, separate from the trial timers.

Storage boundaries. Checkpoint storage sums logical sizes of final registered serving specialists from cache registries and archive-member indices. It includes navigation extension copies, but excludes the shared base, rejected candidates, training intermediates, task data, and logs. E2 extension credentials add no extra model file; its six specialists total 234,776,244 bytes (39,129,374 each), or 223.90 MiB with $1\,\mathrm{MiB} = 2^{20}$ bytes. Names and sizes are listed in `e2_admitted_checkpoint_storage.csv`. Both cached navigation methods have a six-entry caching and admission limit. These sizes are neither measured RAM nor GPU memory.

Table D3: Final registered checkpoint files. Extra storage uses CARVE as the denominator.

	Cache-only		CARVE		Cache-only extra storage vs. CARVE
Experiment	Files	MiB	Files	MiB	(%)
Navigation: caching and admission	5	1.872	4	1.498	25.0
Chip E2	8	298.533	6	223.900	33.3

Extra storage is $100(B_{\mathrm{Cache}}/B_{\mathrm{CARVE}} - 1)$, calculated from exact byte counts. Registered duplicate extension copies are included. The common base, rejected candidates, training intermediates, and logs are excluded. Sizes are logical files, not RAM or GPU memory.

D.2 Paired effects and uncertainty across evaluation seeds

Placement summary columns. Each request's gain is the reduction of its 30-trial median HPWL relative to the matched Frozen median, not classification accuracy. Mean gain and sample SD use all 24 requests, including first arrivals and base fallbacks. Rank is one plus the number of methods with strictly lower median HPWL on that request; ties share a rank. Mean rank must be recomputed when the comparison set changes. A more-than-5%-worse request has median HPWL greater than 1.05 times its matched base median. Raw HPWL is not pooled across circuits.

Navigation effects. For request r, $d_r = 100(\widehat{p}_{\mathrm{CARVE},r} - \widehat{p}_{\mathrm{reference},r})$. Each request uses 100 paired cases, so this equals the success-count difference. The original recomputation covers the complete archive of 420 row files and 42,000 outcomes. The detailed tables use the three displayed streams, matching family, scene parameters, and case range. For each 30-request stream, sample SD is

$$s_d = \sqrt{\frac{1}{29}\sum_{r=1}^{30}(d_r - \bar{d})^2}. \tag{11}$$

SD describes variation across requests sharing models and training history, not independent training seeds or a significance test. The Overwrite and Cache-only contrasts for caching and admission share CARVE observations and are not pooled. The same case index under different scene parameters remains a different case. Tables D4 and D5 retain the displayed stream and family effects.

Table D4: CARVE minus reference, swept-success difference (pp), mean ± sample SD across 30 matched requests. Positive favors CARVE; — means not run.

Stream	Overwrite	Cache-only
Fixed arrivals	8.20 ± 12.47	—
Sampled arrivals	1.40 ± 14.07	—
Caching and admission	1.90 ± 19.40	1.93 ± 10.35

Table D5: Family differences in swept success (pp). Positive favors CARVE. H/T/L counts requests with higher, tied, or lower CARVE success; — means not run. These are the original batches, not the separate-case supplementary comparison.

			CARVE − Overwrite			CARVE − Cache-only		
Stream	Family	Requests	Mean	Range	H/T/L	Mean	Range	H/T/L
Fixed arrivals	F-CO	14	11.57	$[-5, 61]$	11/2/1	—	—	—
Fixed arrivals	F-CS	7	0.86	$[-16, 6]$	6/0/1	—	—	—
Fixed arrivals	F-HO	9	8.67	$[0, 16]$	8/1/0	—	—	—
Sampled arrivals	F-CO	16	5.06	$[-8, 42]$	9/4/3	—	—	—
Sampled arrivals	F-CS	7	−2.86	$[-46, 25]$	4/0/3	—	—	—
Sampled arrivals	F-HO	7	−2.71	$[-14, 9]$	3/1/3	—	—	—
Caching and admission	F-CO	8	−4.50	$[-84, 26]$	4/2/2	4.25	$[-2, 14]$	5/2/1
Caching and admission	F-CS	5	3.60	$[-1, 19]$	1/3/1	0.00	$[0, 0]$	0/5/0
Caching and admission	F-HO	6	6.00	$[-7, 23]$	4/0/2	4.50	$[-15, 49]$	2/1/3
Caching and admission	F-HS	5	13.20	$[-3, 31]$	4/0/1	0.00	$[0, 0]$	0/5/0
Caching and admission	F-SQ	6	−4.50	$[-15, 7]$	1/0/5	−0.50	$[-6, 5]$	2/1/3

Reuse versus own-family training. Ten second/third arrivals of five extension-served families use the same 30 base, own-specialist, and reuse seeds per request (300 matched triples). For method a, let $m_{a,r}$ be its request median and b_r the matched base median. The gain is $g_{a,r} = 100(1-m_{a,r}/b_r)$; the gap is $g_{\mathrm{reuse},r} - g_{\mathrm{own},r}$. Negative gaps mean reuse gains less, even when both improve on the base. Two visits per family are observations, not an interval. Seeds, circuit IDs, and medians are checked against the slot summaries; neither five new circuits nor ten independent training runs are bootstrapped.

Table D6: All ten paired revisits. The gap favors reuse when positive. Lower/equal/higher compare reused HPWL with own-specialist HPWL; they need not order marginal medians.

Slot	Family	Own gain (%)	Reuse gain (%)	Gap (pp)	Lower/equal/higher
5	ibm13	20.71	13.21	−7.50	10/0/20
6	ibm18	17.57	13.27	−4.30	6/0/24
7	bigblue1	9.48	1.96	−7.52	0/0/30
11	ibm13	16.01	6.94	−9.07	8/0/22
12	ibm18	17.32	12.65	−4.67	11/0/19
13	ibm10	9.78	3.03	−6.75	10/0/20
15	bigblue1	15.22	6.13	−9.08	0/0/30
17	ibm15	36.69	29.89	−6.80	13/0/17
19	ibm15	21.65	24.39	2.73	17/0/13
21	ibm10	8.90	7.64	−1.26	13/0/17

Conditional transfer-service intervals. For fixed target c, the selected import and base use the same 30 unique service seeds, disjoint from the target's recorded exam seeds. The estimate is

$$\widehat{g}_c = 100\left(1 - \frac{\mathrm{median}_i\, S_{ci}}{\mathrm{median}_i\, B_{ci}}\right), \tag{12}$$

not the median of individual percentage reductions. Each bootstrap draws 30 indices with replacement and applies them to both vectors before recomputing the medians and ratio. Each target uses 50,000 resamples from NumPy PCG64 seed 20260910; linear interpolation at percentiles 2.5/97.5 gives the interval in Table B5. The seed belongs to analysis, not environment execution.

The earliest-admission selector is held fixed and service is not used to choose a different import. Base routes have zero effect by identity and no specialist interval. Under independent paired draws from that target's generator, the intervals describe evaluation variation on the fixed target/model; they omit training, library selection, and between-circuit uncertainty. They are pointwise, not simultaneous, and crossing zero does not undo an admission on another block.

The analysis inputs in `data/statistics/` retain six stream contrasts, 26 family summaries, ten revisits, 300 matched triples, and six receiver intervals. Original routes, point estimates, and admission decisions remain unchanged.

D.3 Interpreting variation in the recorded results

Request variability, local-training replication, and conditional evaluation uncertainty are distinct. The original navigation streams and E2/E2-X revisits share evolving state; request counts do not

replace independent runs (Agarwal et al., 2021). Recipe repetitions in C.5 repeat local training, not the deployment system. E-T instead conditions on a fixed target and selected import. Its intervals neither repeat Holm admission nor measure selector regret. These variability analyses were added retrospectively; the two intervals containing zero and the limitations of the original caching and admission study remain reported (Cawley & Talbot, 2010).

D.4 Repair triggers on benchmark circuits

Benchmark-circuit repair is triggered by model/family identity. The earlier random-placement diagnostic did not flag those circuits in the pre-execution check; its logged scores are not the evidence triggering repair. The placement studies evaluate training, admission, scope, reuse, and fallback, not a validated OOD detector. Navigation instead triggers repair from measured swept success.

D.5 Recomputation and evidence boundaries

The source archive retains all 420 navigation method–request records, including the additional history in Appendix A.1. The detailed displays use 300 records. Chip medians, gates, and the complete RS trace remain unchanged. File manifests and hashes check consistency of the supplied files, not outcomes of unexecuted methods. The audits do not load policies, tune rules, or issue credentials. Rerunning policies requires the corresponding checkpoints and environments. Per-file timing records remain in the source archive.

E PROOFS AND VALIDATION REQUIREMENTS

The analysis concerns the expected loss on a declared family distribution. It separates the evidence needed to authorize a specialist from the rules that preserve and reuse that specialist. All logarithms below are natural.

E.1 PROOF OF PROPOSITION 1

Fix one repair epoch. For each in-scope family f, the distribution P_f and loss function ℓ_f remain unchanged. The base and each stored specialist retain their complete inference behavior, including any randomization procedure. The expected loss $R_f(M)$ and improvement $\Delta_f(M)$ are defined in Section 2. The probability in Proposition 1 concerns the training and validation history. It does not assert that a specialist beats the base on every sampled instance.

Account for each new credential. Index validation rounds by $k = 1, 2, \ldots$. The history $\mathcal{G}_{k-1}$ contains all information available before round k, including the chosen candidates and their target families. Formally, $\mathcal{G}_{k-1}$ is a sigma-field, meaning a collection of events whose outcomes are known at that time. The allowance α_k is chosen from that history before the new validation data are examined. The event B_k occurs if round k issues at least one credential (j, f) with $\Delta_f(M_j) < 0$. If several pairs are examined together, B_k includes all credentials issued in that round. The epoch starts with no credentials, or includes every initial credential in the same error accounting. After validation stops, take B_k to be the empty event and $\alpha_k = 0$ for unused rounds.

Bound the probability of any false credential. The conditional error assumption gives

$$\Pr(B_k) = \mathbb{E}[\Pr(B_k \mid \mathcal{G}_{k-1})] \leq \mathbb{E}[\alpha_k]. \tag{13}$$

The equality averages the conditional probability over all possible histories. For any finite number N of rounds, the union bound gives

$$\begin{aligned} \Pr\left(\bigcup_{k=1}^{N} B_k\right) &\leq \sum_{k=1}^{N} \Pr(B_k) \\ &\leq \mathbb{E}\left[\sum_{k=1}^{N} \alpha_k\right] \leq \delta. \end{aligned} \tag{14}$$

The last inequality follows from the assumed total allowance $\sum_{k\geq 1} \alpha_k \leq \delta$ on every execution except a set of probability zero. The events on the left grow as N increases, so their probabilities converge to the probability of their union. Therefore,

$$\Pr\left(\bigcup_{k\geq 1} B_k\right) \leq \delta. \tag{15}$$

Neither the union bound nor the averaging step requires independence between validation rounds. Candidates and families may depend on earlier results, provided the conditional error assumption continues to hold.

Apply credential validity to every service decision. On the complementary event in equation 15, every issued credential has nonnegative population improvement. Consider any in-scope service request t. If $g_t = M_0$, its expected loss equals the base loss. Otherwise, the routing rule supplies a credential (j, f_t) for $g_t = M_j$, and hence

$$R_{f_t}(M_0) - R_{f_t}(g_t) = \Delta_{f_t}(g_t) \geq 0. \tag{16}$$

The model, loss, and family distribution remain fixed, so later training cannot alter the population comparison recorded for that credential. The argument holds simultaneously for every service request on the same event of probability at least $1 - \delta$. Out-of-scope requests use M_0 by construction and therefore have exact equality with the base. The service bound in Proposition 1 follows.

The proof identifies why preserving a model and restricting its use to the evaluated family work together. A new credential requires error accounting, whereas another use of an existing credential does not add a new validation event. A change in P_f or inference behavior invalidates the preservation step. A common family name alone does not establish that the distribution remained unchanged.

E.2 Proof of Theorem 2

Fix a target distribution P, a bounded loss ℓ, and a library $M_0, M_1, \ldots, M_J$. Define

$$R_P(M) = \mathbb{E}_{Z\sim P}[\ell(M, Z)], \qquad \Delta_j = R_P(M_0) - R_P(M_j), \qquad \Delta_0 = 0. \tag{17}$$

The validator receives n independent observations from P and the paired losses of all models on those observations. Its target information is limited to that sample, and it may inspect the fixed models. The library and sample size are fixed before the sample is drawn. They may have been chosen using an earlier history, on which this analysis is conditioned. The quantity $n^\star$ is the smallest fixed sample size for which some validator satisfies equation 2 uniformly over all allowed libraries and target distributions.

Estimate population improvement. For candidate j and observation i, set

$$D_{j,i} = \ell(M_0, Z_i) - \ell(M_j, Z_i), \qquad \widehat{\Delta}_j = \frac{1}{n}\sum_{i=1}^{n} D_{j,i}. \tag{18}$$

Both losses lie in $[0, 1]$, so $D_{j,i} \in [-1, 1]$ and $\mathbb{E}[D_{j,i}] = \Delta_j$. For each fixed j, the variables $D_{j,1}, \ldots, D_{j,n}$ are independent because the observations are independent. Paired losses for different models on the same observation may be dependent.

Control all estimation errors. Hoeffding's inequality states that the average of n independent variables in an interval of width $b - a$ satisfies

$$\Pr\big(\big|\overline{X} - \mathbb{E}[\overline{X}]\big| > u\big) \le 2\exp\left(-\frac{2nu^2}{(b-a)^2}\right), \qquad u > 0. \tag{19}$$

Here $\overline{X}$ is their sample average. Applying the inequality with $a = -1$, $b = 1$, and $\varepsilon_n = \sqrt{2\log(2J/\delta)/n}$ gives

$$\Pr\Big(|\widehat{\Delta}_j - \Delta_j| > \varepsilon_n\Big) \le 2\exp(-n\varepsilon_n^2/2) = \frac{\delta}{J}. \tag{20}$$

The union bound over the J candidates therefore gives

$$\Pr(G) \ge 1 - \delta, \qquad G = \bigcap_{j=1}^{J} \Big\{|\widehat{\Delta}_j - \Delta_j| \le \varepsilon_n\Big\}. \tag{21}$$

Dependence between different candidates' estimates does not affect the union bound.

Verify acceptable selection. Approve candidate j when $\widehat{\Delta}_j - \varepsilon_n \ge 0$. Select a passing candidate according to a fixed order, or the base if none passes. On G, every passing candidate satisfies

$$\Delta_j \ge \widehat{\Delta}_j - \varepsilon_n \ge 0. \tag{22}$$

Fallback also has nonnegative improvement because $\Delta_0 = 0$. Thus the selected model meets the first requirement in equation 2 on G.

Recognize an available improvement. Suppose some candidate has $\Delta_j \ge \gamma$. On G, its lower confidence bound satisfies

$$\widehat{\Delta}_j - \varepsilon_n \ge \Delta_j - 2\varepsilon_n \ge \gamma - 2\varepsilon_n. \tag{23}$$

One copy of ε_n accounts for possible underestimation of Δ_j, and the gate subtracts the other copy. The candidate passes whenever $2\varepsilon_n \le \gamma$, which is equivalent to

$$n \ge \frac{8\log(2J/\delta)}{\gamma^2}. \tag{24}$$

At least one specialist is then selected. Both requirements hold on the same event G, proving

$$n^\star \le \left\lceil \frac{8\log(2J/\delta)}{\gamma^2} \right\rceil. \tag{25}$$

Construct fixed models and unknown targets. To prove a worst-case lower bound, it suffices to construct allowed problems that every reliable validator must handle. Let an observation be $Z = (Z_1, \ldots, Z_J) \in \{0,1\}^J$, and fix the losses

$$\ell(M_0, Z) = \frac{1}{2}, \qquad \ell(M_j, Z) = 1 - Z_j, \qquad 1 \le j \le J. \tag{26}$$

Under P_0, all coordinates are independent Bernoulli variables with probability $1/2 - \gamma$ of taking the value one. Under P_j, only coordinate j changes that probability to $1/2 + \gamma$. For a coordinate with probability p of being one, the corresponding specialist has expected loss $1 - p$ and improvement $p - 1/2$. Consequently,

$$\begin{aligned} P_0 &: \quad \Delta_i = -\gamma \quad \text{for } 1 \le i \le J, \\ P_j &: \quad \Delta_j = \gamma, \qquad \Delta_i = -\gamma \quad \text{for } 1 \le i \le J,\ i \ne j. \end{aligned} \tag{27}$$

The subscript on P_0 labels a target distribution, not the base model. All models and loss functions are identical across these problems. Only the unknown target distribution differs, so inspecting the models cannot distinguish the problems. Observing every paired loss reveals the full vector Z and supplies no information beyond that vector.

Identify decisions that must differ. Let $A_j = \{\widehat{j} = j\}$ be the event that the validator selects specialist j. Write $\Pr_i$ for probability when the sample comes from P_i, including any randomization by the validator. Under P_0, all specialists are harmful. Reliable reuse therefore requires

$$\sum_{j=1}^{J} \Pr_0(A_j) \le \delta. \tag{28}$$

The events are disjoint because the validator selects one model. At least one index $j^\star$ satisfies $\Pr_0(A_{j^\star}) \le \delta/J$; otherwise their sum would exceed δ. Under $P_{j^\star}$, the only acceptable specialist is $j^\star$, and its improvement equals γ. Fallback violates the recognition requirement, so

$$p := \Pr_{j^\star}(A_{j^\star}) \ge 1 - \delta, \qquad q := \Pr_0(A_{j^\star}) \le \frac{\delta}{J}. \tag{29}$$

Measure information in the observations. For discrete distributions Q and R, their KL divergence, also called relative entropy, is

$$\mathrm{KL}(Q\|R) = \sum_y Q(y) \log \frac{Q(y)}{R(y)}. \tag{30}$$

Terms with $Q(y) = 0$ contribute zero; a term with $Q(y) > 0$ and $R(y) = 0$ makes the divergence infinite. KL divergence is nonnegative, as follows by applying $-\log u \ge 1 - u$ to $u = R(y)/Q(y)$ and summing with weights $Q(y)$. For independent coordinates, the log likelihood ratio is a sum and its expectation is the sum of their KL divergences. Only coordinate $j^\star$ differs between the two targets. Writing kl for the divergence between Bernoulli distributions gives

$$\begin{aligned} d_\gamma &:= \mathrm{KL}(P_{j^\star}\|P_0) \\ &= \left(\frac{1}{2} + \gamma\right) \log \frac{\frac{1}{2} + \gamma}{\frac{1}{2} - \gamma} + \left(\frac{1}{2} - \gamma\right) \log \frac{\frac{1}{2} - \gamma}{\frac{1}{2} + \gamma} \\ &= 2\gamma \log \frac{1 + 2\gamma}{1 - 2\gamma}. \end{aligned} \tag{31}$$

The same sum argument for n independent observations gives

$$\mathrm{KL}(P_{j^\star}^{\otimes n}\|P_0^{\otimes n}) = n d_\gamma, \tag{32}$$

where $P_i^{\otimes n}$ denotes the distribution of the full sample.

Randomizing the validator does not increase the available information. For a realized sample z, let $w(a \mid z)$ be its probability of selecting index $a \in \{0, \ldots, J\}$. The rule w is the same under both targets. Let Q and R be the respective joint distributions of the sample and selected index. Their likelihood ratio cancels w, and summing over a gives

$$\mathrm{KL}(Q\|R) = \mathrm{KL}(P_{j^\star}^{\otimes n}\|P_0^{\otimes n}) = n d_\gamma. \tag{33}$$

Deterministic selection is the special case where each $w(a \mid z)$ is either zero or one.

Compare observation and decision information. Partition the joint outcomes according to $A = A_{j^\star}$ and its complement A^c. Substituting conditional probabilities into the definition of KL gives

$$\begin{aligned}\mathrm{KL}(Q\|R) = \; & \mathrm{kl}(p,q) + p\,\mathrm{KL}(Q(\cdot \mid A)\|R(\cdot \mid A)) \\ & + (1-p)\,\mathrm{KL}(Q(\cdot \mid A^c)\|R(\cdot \mid A^c)).\end{aligned} \tag{34}$$

Terms with zero weight are interpreted by continuity. Dropping the nonnegative conditional terms yields $nd_\gamma \geq \mathrm{kl}(p,q)$. The inequality is the data-processing inequality for the decision event. Recording only whether the selected index is $j^\star$ cannot increase KL divergence. In this construction every sample vector has positive probability under both targets. Thus $p > 0$ implies $q > 0$, and the logarithms below are well defined.

To bound the binary divergence, define the binary entropy $h(p) = -p\log p - (1-p)\log(1-p)$. Differentiation gives its maximum $\log 2$ at $p = 1/2$. Rearranging the binary KL formula gives

$$\begin{aligned}\mathrm{kl}(p,q) &= p\log(1/q) + (1-p)\log(1/(1-q)) - h(p) \\ &\geq p\log(1/q) - \log 2 \\ &\geq (1-\delta)\log(J/\delta) - \log 2.\end{aligned} \tag{35}$$

The second term in the first line is nonnegative, and the final line uses equation 29. Since $\delta \leq 1/4$ and $J \geq 1$, we have $1 - \delta \geq 3/4$ and $\log(J/\delta) \geq \log 4 = 2\log 2$. Therefore,

$$nd_\gamma \geq \frac{1}{4}\log(J/\delta). \tag{36}$$

Convert information into sample size. For $0 < \gamma \leq 1/4$, the inequality $\log(1+u) \leq u$ gives

$$\begin{aligned}d_\gamma &= 2\gamma\log\left(1 + \frac{4\gamma}{1-2\gamma}\right) \\ &\leq \frac{8\gamma^2}{1-2\gamma} \leq 16\gamma^2.\end{aligned} \tag{37}$$

Combining with equation 36 yields

$$n^\star \geq \frac{\log(J/\delta)}{64\gamma^2}. \tag{38}$$

Together, equations 25 and 38 prove Theorem 2. The bounds have the same dependence on library size, confidence, and detectable improvement. They are worst-case bounds over bounded-loss problems, so they do not assert that every particular library requires that many examples.

E.3 A VALIDATION RULE ACROSS SUCCESSIVE REQUESTS

The upper-bound construction in Theorem 2 supplies the conditional error control required by Proposition 1. Its proof applies to any error tolerance in $(0,1)$. Fix an epoch allowance $0 < \delta < 1$ and consider a round k that evaluates $J_k \geq 1$ fixed candidates on one target family. Choose a desired detectable improvement $0 < \gamma_k \leq 1/4$ before sampling, and allocate

$$\alpha_k = \frac{\delta}{k(k+1)}, \qquad n_k = \left\lceil \frac{8}{\gamma_k^2}\log\frac{2J_k k(k+1)}{\delta} \right\rceil. \tag{39}$$

The identity $1/(k(k+1)) = 1/k - 1/(k+1)$ gives $\sum_{k\geq 1}\alpha_k = \delta$. Conditional on the complete prior history, draw n_k fresh independent examples from the declared target distribution. Use the confidence-bound gate in Appendix E.2 with $J = J_k$ and $\delta = \alpha_k$. The upper-bound proof controls every passing candidate at once, so $\Pr(B_k \mid \mathcal{G}_{k-1}) \leq \alpha_k$. It also ensures that a specialist is selected whenever an available candidate improves by at least γ_k, on the same confidence event. A round that covers several families must include all issued credentials in its conditional error bound, for example by allocating separate allowances to the family evaluations.

The sample size in equation 39 is sufficient for recognition as well as acceptable selection. The confidence-bound gate controls false approval for any fixed positive sample size, but a smaller sample may cause fallback even when useful expertise exists. Evaluating all models on n_k observations can require $(J_k + 1)n_k$ model executions. The theorem therefore does not provide a logarithmic bound on runtime or guarantee that validation is cheaper than a particular training procedure.

Relationship to the recorded experiments. The experimental contracts in Appendix A.6 implement the same separation between retention, admission, and reuse, but use different statistical gates. Navigation requires an observed gain of at least 15 successes on 100 paired cases. Placement combines a reduction in median HPWL with a signed-rank test, with multiplicity handling for extension exams. Neither rule is the bounded-mean confidence gate analyzed above. An application to HPWL would need a fixed bounded loss and a test for its expected difference. Improvement in a median does not by itself establish improvement in an expectation.

The sampling conditions also require explicit verification. A navigation family includes settings such as scene size and pedestrian count (Appendix A.7). Its P_f must specify their distribution together with episode randomness. Evaluation at one setting does not establish the guarantee for other settings with the same family key. Appendix C.2 records overlap between the permitted training and admission index ranges in the original navigation runs. The overlap does not prove training exposure, but disjointness and conditional independence have not been established by that record. A retrospective service subset cannot establish validity of the original admission decision. Finally, correction within an extension exam does not by itself allocate error across all exams in an epoch. The results above give conditions for a population guarantee; they do not retrospectively certify the empirical gates or promise the same realized gain on each later request.

F RECORDED MACRO-PLACEMENT EXAMPLES

Figures F1 and F2 show archived rollouts for the eight circuits in the revisit study. Each row compares Random, the frozen base, the final shared Overwrite checkpoint, and a saved CARVE specialist or candidate using the same recorded seed. Each row gives the circuit's total loaded macro count; this is not the number of rectangles displayed. The macro inventory shows exported macro shapes in a schematic arrangement, rather than an initial placement. Placement panels share the circuit's canvas scale; colors encode log macro area and identify the same macro across methods, while gray marks fixed nodes and ports. Overwrite uses the same final checkpoint throughout, rather than the successive checkpoints used to serve the request stream.

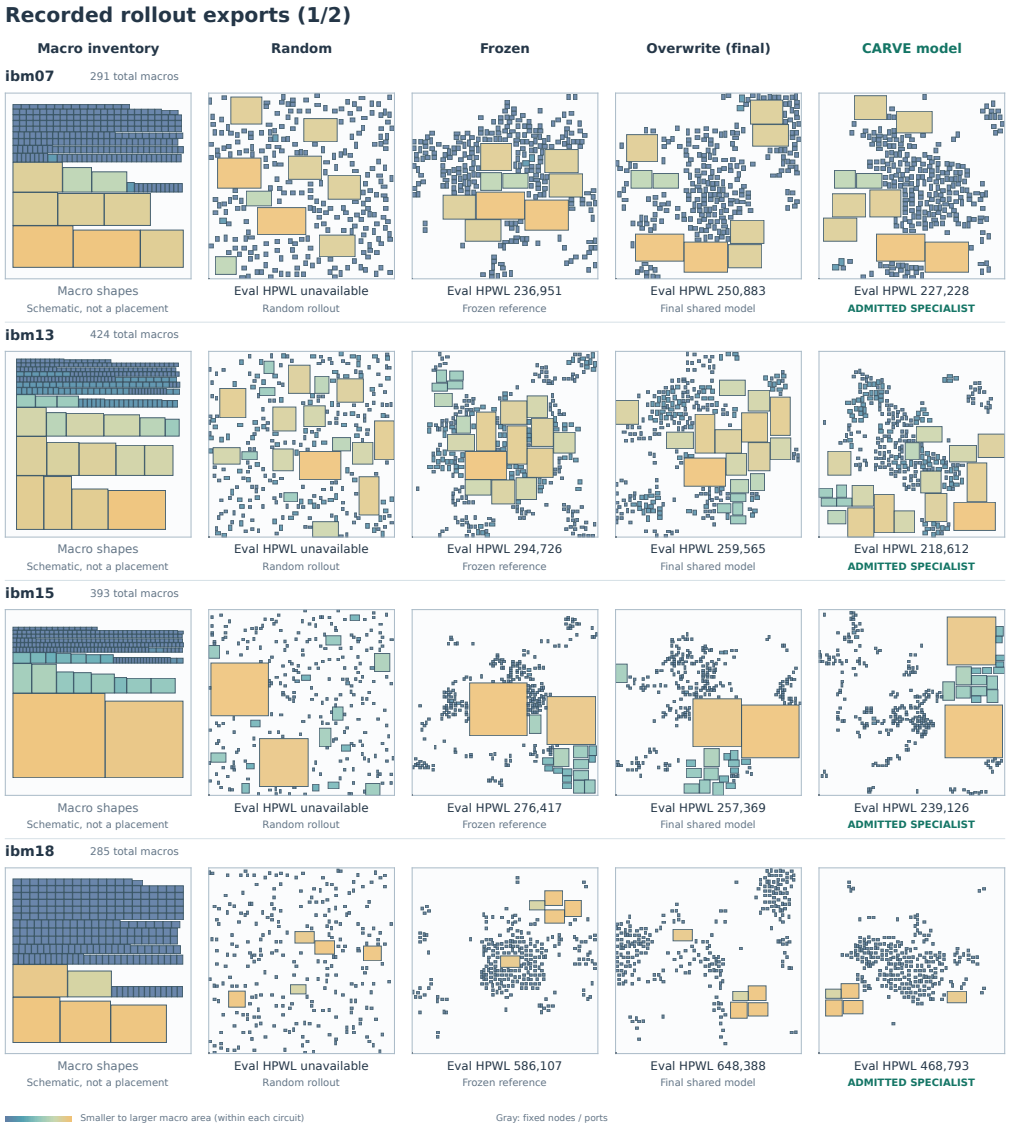


Figure F1: Recorded rollouts for ibm07, ibm13, ibm15, and ibm18. Row labels give circuit totals; panels show exported subsets. Eval HPWL comes from scoring records. The CARVE column shows admitted specialists.

The adaptec3 row reports the same circuit total as Figure 3. The panels show an exported subset from a different run. Only the bigblue1 model exports include all loaded macros. The archived evaluator continues the initial rollout for adaptec2 and adaptec3 before computing HPWL; IBM scoring remains limited to the placed subset.

Eval HPWL comes from the scoring records, not a recomputation from the displayed exports. Table A10 reports 30-trial medians. Admission labels come from separate evaluations. The two adaptec candidates were rejected and not deployed. Their Random exports are incomplete and have no evaluation HPWL.

Recorded overlaps are retained and shapes are clipped at the canvas. A full rollout means that all loaded macros are represented; it does not establish legality. The archived evaluator has no final overlap or boundary check. The reported HPWL gains therefore describe the archived evaluation protocol, and full-placement legality remains unverified.

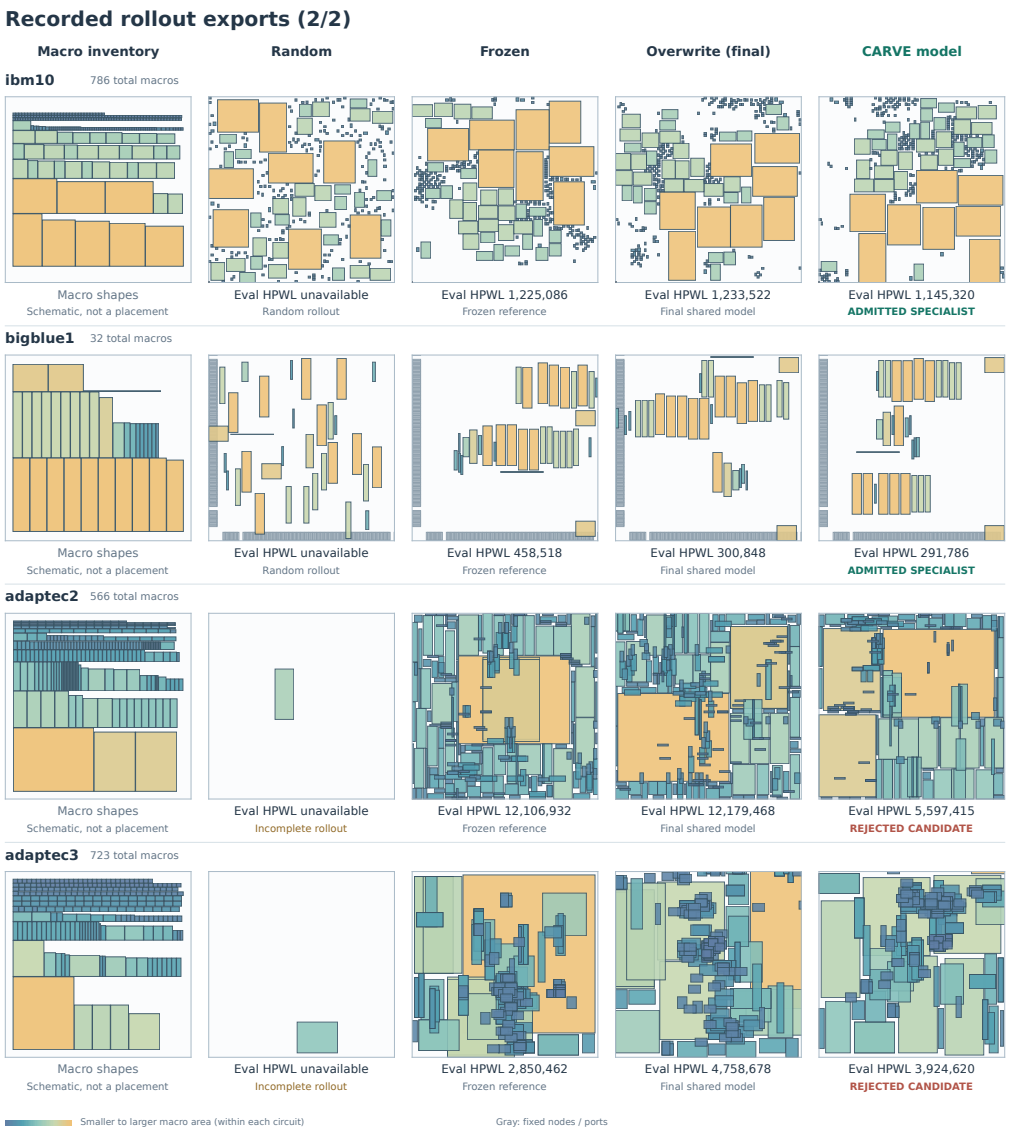


Figure F2: Recorded rollouts for ibm10, bigblue1, adaptec2, and adaptec3. Row labels give circuit totals. The ibm10 and bigblue1 models are admitted specialists; the two adaptec models are rejected candidates shown for inspection. The incomplete Random rollouts are explicitly marked.