Curriculum Learning for Qubit Mapping Across Hardware Topologies

Aleksandr Govenko[1]

Supervisor(s): Sebastian Feld[1]**, Akash Kundu**[1]**, Matthijs Spaan**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Aleksandr Govenko
Final project course: CSE3000 Research Project
Thesis committee: Matthijs Spaan, Sebastian Feld, Anna Lukina

Abstract

Compiling quantum circuits for physical hardware requires an initial mapping step that assigns virtual qubits to physical qubits such that interacting pairs are placed on connected hardware locations. Current approaches train a separate agent per device topology, requiring significant compute for each new hardware generation and transferring no knowledge across devices. This work investigates whether curriculum learning — progressively training a reinforcement learning agent on hardware topologies of increasing size — can produce a single agent that generalises to unseen topologies. We evaluate three curriculum variants differing in replay ratio and warmup length, alongside three non-curriculum baselines, in the QGym `InitialMapping` environment using MaskablePPO. Results show that curriculum agents outperform single-topology and single-size training on held-out topologies, reaching strong frontier performance with greater sample efficiency than direct training. Against unordered exposure to the same topology distribution, however, curriculum ordering's advantage holds on the target topology size but not on generalisation to unseen topologies. While absolute performance remains modest and variance across seeds is substantial, the findings establish curriculum learning as a viable approach to topology-general qubit mapping and provide a proof of concept for training a single model that transfers across hardware topologies, reducing the computational cost of re-training for each new device.

1 Introduction

Quantum computers promise to solve certain classes of problems that are infeasible for classical hardware — simulating quantum chemistry, breaking cryptographic schemes, or optimising large combinatorial structures — but realising this potential in the near term demands careful engineering [14]. Today's quantum devices are *noisy*: every gate operation has a non-negligible probability of error, and quantum states decohere rapidly. The qubits in these devices are also *sparsely connected*: not every qubit can interact directly with every other. Most quantum algorithms assume all-to-all qubit connectivity, so when interacting qubits end up on non-adjacent hardware nodes, the compiler must insert extra operations to bridge the gap — making the program slower and less reliable. Running a computation reliably on such hardware requires a compilation step that maps abstract programs onto these physical constraints.

Initial qubit mapping — the first and critical pass of this compilation pipeline [8][19] — is the focus of this work. The problem is analogous to register allocation in a classical compiler: virtual qubits play the role of program variables, and physical qubits act as a fixed set of hardware "registers" with constrained connectivity. Just as a register allocator must assign variables to registers so that simultaneously live variables do not conflict, the qubit mapper must assign virtual qubits to physical qubits so that pairs that interact in the circuit are physically adjacent on the device. When they are not, the compiler must insert *SWAP gates* — sequences of operations that physically relocate quantum information along available connections. Unlike classical spills, which just incur a memory access penalty, each SWAP gate introduces additional noise and increases circuit depth, degrading the fidelity of the entire computation.

Current compilers rely on hand-engineered heuristics for initial mapping — greedy graph matching [19], look-ahead strategies [8], simulated annealing [22] — that must be retuned for every new device architecture. Reinforcement learning (RL) offers a data-driven alternative: an agent learns a mapping policy by interacting with a compilation environment and receiving a reward signal proportional to mapping quality [20]. Prior work has demonstrated that RL agents can produce viable mappings on individual device topologies [12], but existing approaches train a *separate* agent for every target topology. No knowledge transfers across hardware generations, and cross-topology generalisation has not been studied in this setting.

Curriculum learning — progressively exposing an agent to tasks of increasing difficulty — is the approach we investigate to close this gap [11]. Representations acquired on simpler topologies may serve as structural priors that accelerate learning on larger, more complex devices. To our knowledge, curriculum learning has not previously been applied to the initial mapping phase of quantum circuit compilation. We investigate three curriculum variants differing in replay ratio and warmup length, alongside three non-curriculum baselines (fixed topology, fixed size, and random sampling), in the QGym `InitialMapping` environment [21] using MaskablePPO, evaluating whether a single trained agent can generalise zero-shot to unseen hardware topologies.

This work is guided by the following central research question:

> *To what extent can curriculum learning over hardware topologies produce an RL agent for initial qubit mapping that generalises to unseen devices?*

Three sub-questions operationalise this:

SQ1 How does an agent trained on a single fixed topology perform when evaluated on unseen topologies of varying size and structure?

SQ2 To what extent does a topology curriculum improve zero-shot performance on held-out topologies compared to single-topology training?

SQ3 How does curriculum training affect sample efficiency relative to training directly on large topologies?

The remainder of this paper is organised as follows. Section 2 formalises the initial mapping problem, surveys reinforcement learning approaches to quantum compilation, and introduces curriculum learning — identifying the gap that this work addresses. Section 3 describes the environment, agent architecture, the training strategies, and the evaluation protocol. Section 4 presents experimental results across training and held-out topologies. Section 5 discusses the implications

of these results, limitations, and directions for future work. Section 6 concludes. Section 7 discusses responsible research considerations.

2 Background and Related Work

2.1 Quantum compilation and the initial mapping problem

In gate-based quantum computing, a quantum program is expressed as a circuit operating over virtual qubits. Before execution on physical hardware, this circuit must be compiled — transformed into a sequence of operations that respects the connectivity and gate constraints of a specific device. In the current Noisy Intermediate-Scale Quantum (NISQ) era, where devices are limited in qubit count and prone to gate errors, the quality of compilation directly determines whether a circuit can be executed with meaningful fidelity [14]. The compilation pipeline consists of three passes that are each NP-hard: initial mapping, routing, and scheduling [21].

Initial mapping is a critical first pass. It establishes a bijection $f : V \to P$ between the virtual qubits of a circuit's interaction graph $G_I = (V, E_I)$ and the physical qubits of a device's coupling graph $G_C = (P, E_C)$. The interaction graph encodes the two-qubit gate structure of the circuit: vertices are virtual qubits and edges connect pairs that interact. The coupling graph encodes the hardware: vertices are physical qubits and edges represent supported connections. While different metrics can quantify mapping quality, the most widely adopted one, which we follow, is to minimise the number of mismatched edges $E_M \setminus E_C$, where $E_M = \{(f(v), f(u)) : (v, u) \in E_I\}$ — that is, interactions present in the mapped circuit but unsupported by the hardware [21][19][8].

Unsatisfied interactions must be resolved in the subsequent routing pass. On superconducting architectures, where qubit connectivity is sparse and fixed, this is achieved by inserting SWAP gates that physically relocate quantum information along available edges [3]. Each additional SWAP increases circuit depth and introduces further noise, since every gate operation is a potential error source and qubits have limited decoherence times. The initial mapping problem is a generalisation of the subgraph isomorphism problem — finding a placement of the interaction graph within the coupling graph such that as many edges as possible coincide — and is NP-hard [19]; routing is NP-hard independently [6].

Current compilers address initial mapping with hand-engineered heuristics — greedy graph matching [19], look-ahead methods [8], and simulated annealing [22] — that must be manually re-tuned for each new device and cannot transfer knowledge across hardware generations.

2.2 Reinforcement learning for initial mapping

Reinforcement learning (RL) offers an alternative: an agent interacts with a compilation environment by iteratively placing virtual qubits onto physical qubits, one per step, and receives a reward signal that guides it towards high-quality mappings [20]. The reward is naturally sparse — the edge overlap can only be computed once all qubits are placed — making policy gradient methods a natural fit. Note that this metric is itself a proxy for the quality of the final compiled program. QGym provides Gymnasium-compatible environments for all three compilation passes [21]; in the `InitialMapping` environment, the coupling graph is fixed across episodes while a new interaction graph is sampled each episode, allowing an agent to learn a policy for a given hardware topology. Policy implementations are accessible through Stable-Baselines3 [17], a widely-used library compatible with these environments.

Prior work has explored RL for both initial mapping and routing [5][9]. Oancea et al. [12] provide the most systematic study of this regime within QGym, training and evaluating policy gradient agents across IBM, Rigetti, and Google topologies ranging from 5 to 127 qubits. Their results show that agents can produce near-optimal mappings on small architectures, but performance degrades markedly at larger scales: on 20-qubit devices some algorithms fail to converge entirely, and by 27 qubits even the best-performing agents only marginally outperform a random mapper. Crucially, existing approaches — including those of Oancea et al. — train a separate agent for every target topology; the input dimensionality of the multi-layer perceptron (MLP) policy is fixed to the qubit count of a single device at initialisation, so no knowledge transfers across hardware generations.

2.3 Curriculum learning in reinforcement learning

Curriculum learning is a training strategy in which an agent is exposed to tasks of increasing difficulty, with the expectation that knowledge acquired on simpler tasks accelerates learning and improves sample efficiency on harder ones [1]. Narvekar et al. [11] formalise this for RL: a curriculum is a directed acyclic graph over tasks, where each edge encodes a training precedence, and the simplest common case is a linear sequence $[m_1, m_2, \ldots, m_f]$ ordered by increasing complexity. The method has been shown to improve sample efficiency, and in some settings final performance, across a range of domains. The relative weighting between newly available and previously encountered experience has also been studied in adjacent settings: Rolnick et al. [18] mix on-policy learning on novel tasks with off-policy replay of past tasks in continual reinforcement learning, finding that a roughly even split between new and replayed experience is effective, though final performance is not highly sensitive to the exact ratio.

The success of curriculum learning depends on whether representations learned on simpler tasks transfer to more complex ones. In our setting, the relevant notion of difficulty is topology size and structural complexity: a small line topology imposes far simpler mapping constraints than a larger one or a real multi-dimensional device graph. We hypothesise that a curriculum over topologies of increasing size provides a natural ordering the agent can exploit; structural irregularity or a learned difficulty estimate are plausible alternative orderings we do not explore here. Because MLP policies have a fixed observation size, this requires some way of handling topologies of varying qubit count within a single architecture; we address this via padding, detailed alongside the observation space in Section 3.3.

3 Methodology

This section describes the compilation task, the curriculum strategies investigated, the reinforcement learning environment and agent, and the evaluation protocol.

3.1 Problem Formulation

We study the *initial qubit mapping* problem: given a quantum circuit expressed over virtual qubits and a target hardware device, find a bijection $f : V \to P$ from the virtual qubits V of the circuit's interaction graph $G_I = (V, E_I)$ to the physical qubits P of the device's coupling graph $G_C = (P, E_C)$ that minimises the number of unsatisfied interactions. An interaction $(v, u) \in E_I$ is *satisfied* if the corresponding physical pair $(f(v), f(u)) \in E_C$; unsatisfied interactions must be resolved by the routing pass through additional SWAP gates.

We evaluate mapping quality using a normalised reward $\tilde{r}$ derived from the raw episode reward (Equation 2) for comparison and evaluation purposes only; training runs on the raw reward signal directly:

$$\tilde{r} = \frac{|E_{\text{good}}|}{|E_I|} - \frac{|E_{\text{bad}}|}{5\,|E_I|}, \tag{1}$$

where E_{good} is the set of satisfied interactions and E_{bad} the set of unsatisfied ones. Normalising by $|E_I|$ makes $\tilde{r}$ comparable across circuits and topologies of different sizes. A perfect mapping achieves $\tilde{r} = 1$; a mapping with no satisfied edges achieves $\tilde{r} \leq 0$.

3.2 Training Strategies

We investigate whether structured exposure to topologies of increasing size improves generalisation over simpler training regimes. Training topologies are predominantly grid-like graphs with 4 to 8 qubits, motivated by real hardware: Rigetti's square-lattice devices and Google's Sycamore and Bristlecone processors are grid-structured, and IBM's heavy-hex lattice is itself derived from a square grid [2]; line graphs are a degenerate 1×n case of the same family. To increase topology diversity at smaller qubit counts, we additionally include T-shaped topologies. We compare three non-curriculum baselines against a family of curriculum variants.

Baselines

Fixed topology. The agent trains exclusively on a single fixed graph, `line_8`. This replicates the standard single-topology setting of prior work [12].

Fixed size. The agent trains on all 8-qubit topologies, with a new topology sampled uniformly at random each episode. This isolates the effect of topology variety at a fixed difficulty level.

Random curriculum. At the start of each episode, a topology is sampled uniformly at random from the full set of 4- to 8-qubit training topologies. This provides an unordered multi-topology baseline for comparison with the curriculum variants.

Curriculum Variants

All curriculum variants share the same progression mechanism: topologies are ordered by increasing size, and the agent advances to the next once its mean normalised reward on the current topology exceeds $\tilde{r} = 0.5$, meaning about 58.3 of all interactions are correctly mapped. Progress is checked every 5 000 training steps, and advancement requires three consecutive evaluations above the threshold, so an agent spends at least 15 000 steps on a topology before advancing, even in the best case. The threshold $\tilde{r} = 0.5$ is a design choice. While individual circuit-topology pairs may not achieve it, it is reachable in expectation over the randomly sampled interaction graphs used in training. We decompose the curriculum design into two independent axes.

Sampling ratio. Once the agent advances to a new frontier topology, training draws from two pools: the *frontier pool*, containing only the newly unlocked topology, and the *history pool*, containing all previously mastered topologies. The sampling ratio controls the split between these two pools. We investigate three ratios (frontier : history): 100:0, 80:20, and 50:50. The 100:0 variant trains exclusively on the frontier until the next advancement, providing maximum focus on new material; the 50:50 variant balances new learning with continued rehearsal of earlier topologies. Once the agent has advanced past the largest topology (8 qubits), training continues for the remainder of the 1 000 000 timesteps using the same sampling ratio, with the 8-qubit topology remaining the frontier and no further advancement occurring.

Warmup. As an independent axis, a warmup phase is applied after each advancement. During warmup, the agent trains exclusively on the frontier topology for a fixed number of steps before the sampling ratio above takes effect. This gives the agent time to form stable representations on the new topology before revisiting earlier ones. We investigate four warmup lengths: 0 (no warmup), 10 000, 30 000, and 50 000 steps, with the 0-step condition serving as a control that isolates the effect of the sampling ratio alone.

Since warmup length is immaterial when no historical replay occurs, the 100:0 ratio is evaluated at a single representative warmup length; the 80:20 and 50:50 ratios are each evaluated at all four warmup lengths. This yields nine curriculum configurations in total.

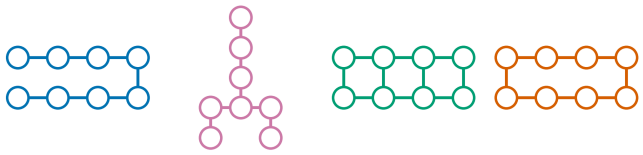

Figure 1: Representative coupling graphs used in this work, shown at 8 qubits: a line graph (`line_8`), a T-shaped tree, a grid (`grid_2x4`), and a cycle (`cycle_8`). Training topologies are predominantly grid-like graphs with 4 to 8 qubits, supplemented with T-shaped topologies for additional diversity at smaller sizes.

3.3 Environment

Experiments are conducted in the `InitialMapping` environment from QGym [21], a Gymnasium-compatible suite for training RL agents on quantum compilation tasks. At the start of each episode, the environment samples a random interaction graph G_I while keeping the coupling graph G_C fixed. The agent then places virtual qubits onto physical qubits one at a time until all virtual qubits are assigned.

Observation space. The observation is a dictionary with three components, all padded to a fixed maximum size $n_{\max}$:

- **Mapping vector** (length $n_{\max}$): each entry records the virtual qubit assigned to the corresponding physical slot; unmapped active slots are encoded as n_{nodes} and inactive padding slots as $n_{\max}$.
- **Interaction matrix** ($n_{\max} \times n_{\max}$, flattened): the adjacency matrix of G_I, zero-padded to $n_{\max}^2$ entries.
- **Connection matrix** ($n_{\max} \times n_{\max}$, flattened): the adjacency matrix of G_C, zero-padded to $n_{\max}^2$ entries.

With $n_{\max} = 8$, the raw observation comprises 8 discrete mapping entries and 128 binary matrix entries. After preprocessing by Stable-Baselines3's `MultiInputPolicy` — which one-hot encodes the mapping vector over $n_{\max} + 1$ values — the concatenated feature vector has dimension $8 \times 9 + 64 + 64 = 200$.

Action space. At each step the agent selects a pair (physical qubit, logical qubit) from a $\texttt{MultiDiscrete}([n_{\max}, n_{\max}])$ space. An action mask is applied at every step, restricting valid choices to physical and logical qubits that are both within the current topology size and not yet assigned. The mask is factorised: legal physical choices and legal logical choices are each represented as a binary vector of length $n_{\max}$, giving a combined mask of shape $2 \times n_{\max}$.

Reward function. The environment uses QGym's `EpisodeRewarder`, which issues a sparse, episode-level reward. Legal non-final steps receive reward 0. Upon completing a full mapping, the agent receives

$$r = 5\,|E_{\text{good}}| - |E_{\text{bad}}|. \tag{2}$$

The coefficient 5 follows QGym's default reward design.

3.4 Agent

We use `MaskablePPO` [16] with a `MultiInputPolicy`, following the experimental setup of Oancea et al. [12], who show that action masking is essential in this setting: standard PPO fails to learn a competitive policy, while MaskablePPO converges reliably. Action masking eliminates invalid placements entirely, so no penalty for illegal actions is required.

Table 1 summarises the network and optimisation configuration, all of which follow Stable-Baselines3 defaults and were not tuned; the policy head maps the shared trunk to $2\,n_{\max} = 16$ action-mask logits, and the value head to a scalar.

Table 1: Agent configuration. Network and optimisation hyperparameters follow Stable-Baselines3 defaults except where noted, and were not tuned.

Parameter	Value
Policy/value trunk	MLP, hidden dims $[64, 64]$, Tanh
Learning rate	3×10^{-4}
Discount factor γ	0.99
Rollout length n_{steps}	512
Mini-batch size	64
Epochs per update	10

3.5 Evaluation Protocol

All agents are evaluated on a fixed set of topologies after training, with no further learning. Evaluation topologies are grouped into two sets:

- **Training topologies**: grid-like graphs with 4 to 8 qubits, supplemented with T-shaped topologies for additional diversity at smaller sizes, matching the training distribution. These assess in-distribution performance.
- **Held-out topologies**: cycle graphs (`cycle_6`, `cycle_8`) and additional grid configurations (`grid_2x3`, `grid_2x4`), never seen during training. These topologies exist at 6 and 8 qubits only, so held-out evaluation does not extend across the full 4–8 qubit range covered by training topologies. These assess zero-shot generalisation to unseen topologies.

The primary metric is mean normalised reward $\tilde{r}$ (Equation 1), reported per topology as mean and standard deviation across 10 independent random seeds.

4 Experiments and Results

All agents are trained for $1\,000\,000$ timesteps using 10 independent random seeds, and evaluated according to the protocol in Section 3.5: results are reported as mean normalised reward $\tilde{r}$ with standard deviation across seeds.

4.1 Curriculum vs. Baseline Comparison

Table 2 summarises the performance of three curriculum variants and the non-curriculum baselines across two evaluation slices: held-out topology performance and all 8-qubit topologies (q8).

All three curriculum variants outperform the narrow-training baselines (q8-only, fixed-line8) on held-out topologies, consistent with the expectation that exposure to multiple topology sizes during training aids generalisation. On q8, this holds for the low-replay curricula (100/0, 80/20); 50/50 (0.369) falls below both fixed-line8 (0.428) and q8-only (0.384), a consequence of the replay-ratio tradeoff discussed below, where heavy replay of smaller topologies costs specialisation on the largest target topology.

Random-curriculum sees the same topology diversity as the ordered curricula, but without any size-based ordering; comparing it against them isolates whether curriculum learning's benefit comes from exposure to multiple topology sizes alone, or specifically from the order in which

Table 2: Mean normalised reward $\tilde{r}$ ($\pm$ std, 10 seeds) for curriculum variants and baselines. Held-out: mean over `cycle_6`, `cycle_8`, `grid_2x3`, `grid_2x4`. Q8: mean over all 8-qubit evaluation topologies.

Condition	Held-out	Q8
100/0, 30k warmup	0.467 ± 0.035	0.462 ± 0.036
80/20, 30k warmup	0.502 ± 0.032	0.432 ± 0.024
50/50, 30k warmup	0.494 ± 0.031	0.369 ± 0.037
q8-only baseline	0.389 ± 0.046	0.384 ± 0.043
fixed-line8 baseline	0.421 ± 0.043	0.428 ± 0.043
random curriculum	0.487 ± 0.032	0.315 ± 0.038

they are presented. On held-out topologies (Section 3.5), random-curriculum (0.487) is broadly competitive with the ordered curricula, narrowly exceeding 100/0 (0.467) and trailing 80/20 (0.502) and 50/50 (0.494) by 0.015 and 0.007 respectively — both well within the standard deviations reported in Table 2 (0.031–0.032 for these three conditions). On q8 the comparison is more conclusive: this metric is averaged over many more training-distribution topologies than the four-topology held-out set, making it a more statistically reliable comparison, and on it random-curriculum (0.315) is the weakest of all six conditions by a wide margin — 0.054 below even the next-lowest condition (50/50, at 0.369). This asymmetry suggests ordering matters for frontier (q8) performance specifically; for generalisation to structurally novel topologies, the evidence does not distinguish curriculum ordering from simple topology diversity. We examine the mechanism behind the q8 effect further in Section 4.2.

Within the ordered curricula, replay ratio produces a tradeoff: held-out generalisation improves with replay (100/0: 0.467, 80/20: 0.502, 50/50: 0.494, with 80/20 and 50/50 close enough to be within noise of each other), while q8 performance is highest for 100/0 (0.462) and declines with increasing replay (80/20: 0.432, 50/50: 0.369). Frequent replay of smaller topologies aids broad generalisation at some cost to specialisation on the largest target topology — enough cost, in 50/50's case, to fall below both narrow-training baselines on q8 specifically, as noted above.

The fixed-line8 baseline outperforms q8-only on both metrics despite training on a single fixed topology, suggesting that even one well-learned 8-qubit graph transfers somewhat better than exposure to varied 8-qubit structures without any smaller-topology grounding. On held-out topologies, both narrow-training baselines remain below every multi-size training condition, including random-curriculum.

Some pairwise comparisons remain within one standard deviation of each other — notably 80/20 versus 50/50 on held-out topologies, 80/20 versus fixed-line8 on q8 (0.432 versus 0.428), and fixed-line8 versus q8-only on both metrics — and should be read as trends rather than definitive rankings.

4.2 Curriculum Initialization Effect

Figure 2 compares q8 evaluation reward over the full training run across all five conditions. Because curriculum variants

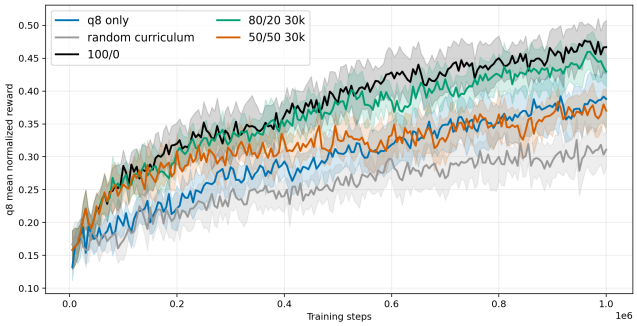

Figure 2: Training curves of q8 evaluation reward over timesteps, comparing the q8-only and random-curriculum baselines against three curriculum variants (100/0, 80/20 with 30k warmup, 50/50 with 30k warmup). Shaded regions indicate standard deviation across seeds.

spend part of their training budget on smaller topologies before reaching q8, they accumulate fewer q8-specific training steps than q8-only for the same total budget, putting any curriculum condition that outperforms q8-only on this axis at a disadvantage.

100/0 and 80/20 converge fastest and reach the highest reward throughout training despite this disadvantage, indicating positive transfer from the smaller topologies seen earlier in the curriculum. The 50/50 variant trails both throughout most of training, converging to an asymptote close to q8-only's only by the end of the 1 000 000-step budget. The reason is dosage: 50/50 has the largest deficit of the three curricula, spending at most half of its post-advancement steps on q8 itself, while q8-only spends its entire budget there. Reaching comparable performance with substantially fewer q8-specific steps is itself a sign of transfer: the deficit leaves little room for transfer to additionally show up as an advantage over q8-only. The random-curriculum baseline, which samples uniformly across all topology sizes throughout training with no recency bias toward q8, remains the weakest condition for the entire run.

4.3 Replay and Warmup Matrix

Evaluation reward declines with topology size for nearly every condition, consistent with the general difficulty scaling reported by Oancea et al. [12]. The exception is q8: conditions with concentrated late-stage training exposure to q8 — fixed-line8, q8-only, and the low-replay curricula — show a partial recovery at q8 relative to q7, while 50/50 and random-curriculum, which dilute or eliminate q8-specific recency, continue the downward trend through q8. We return to this pattern in the Discussion.

The replay-ratio tradeoff extends across topology sizes: generalisation to smaller, unseen-during-training-at-this-ratio sizes improves with replay, while q8-specific performance is generally weaker for 50/50 than for 100/0 and 80/20, mirroring the held-out-versus-frontier tradeoff observed in Table 2.

Within a fixed replay ratio, the four warmup lengths (0, 10k, 30k, 50k) produce final scores within a narrow band at every topology size — for example, 0.653–0.670 across

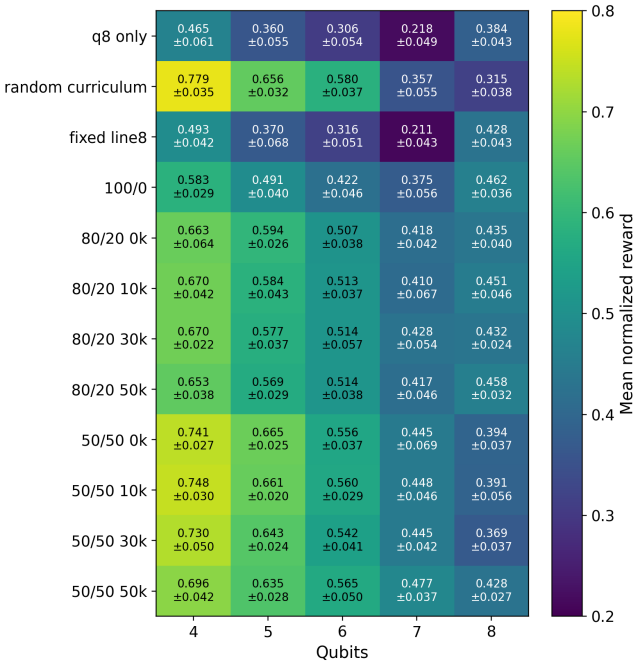

Figure 3: Mean normalised reward $\tilde{r}$ across topology size (4–8 qubits) for all baselines and curriculum configurations (mean $\pm$ std, 10 seeds). Curriculum variants are evaluated across four warmup lengths (0, 10k, 30k, 50k) at each replay ratio, except 100/0, for which warmup is immaterial and a single representative configuration is shown.

all four 80/20 warmup conditions at q4, and 0.432–0.458 at q8 — indicating that warmup length has little effect on final performance. This extends to the 0-step (no-warmup) condition, ruling out warmup as a meaningfully contributing factor within the range tested. The same insensitivity holds for held-out generalisation: across the four 80/20 warmup conditions, held-out scores range from 0.502 to 0.518, and across the four 50/50 conditions from 0.494 to 0.528.

The 100/0 condition serves as a no-replay control: since replay probability is zero, it is evaluated at a single representative warmup setting rather than all four, as warmup is immaterial when there is no historical pool to warm up before revisiting.

5 Discussion

5.1 Curriculum Learning and Zero-Shot Generalisation

Curriculum ordering's benefit is concentrated on frontier performance. Section 4.1 shows a consistent advantage over both narrow-training baselines and random-curriculum on q8, but no advantage over random-curriculum on held-out generalisation beyond what topology diversity alone provides. This points to curriculum learning's contribution here being specialisation efficiency on the target topology family more than broader structural generalisation.

The training curves in Section 4.2 reinforce the frontier finding specifically. Curriculum variants face a structural q8-dosage disadvantage, since part of their training budget is spent on smaller topologies before reaching q8 at all. That 100/0 and 80/20 nonetheless converge faster and higher than q8-only despite this disadvantage, and that even 50/50 — with the largest such deficit — still reaches a comparable asymptote, indicates transfer from smaller topologies: the curricula reach this asymptote with fewer q8-specific steps than q8-only used for its entire budget.

5.2 Replay and the Generalisation–Specialisation Tradeoff

The heatmap in Section 4.3 reframes the apparent difficulty of q7 specifically. Across every condition, q7 is the largest size with no dedicated recent training exposure; q8, by contrast, benefits from recency for conditions whose late-stage training concentrates there. The partial recovery seen at q8 for fixed-line8, q8-only, and the low-replay curricula is therefore a recency effect, not a sign that q7 is structurally harder than q8. 50/50 and random-curriculum, which dilute or eliminate that recency, show no such recovery.

The replay-ratio tradeoff established in Sections 4.1 and 4.3 contrasts with findings in continual reinforcement learning, where Rolnick et al. [18] report that the replay ratio in their CLEAR method has little effect on performance. Their replay mechanism differs from ours: CLEAR applies off-policy learning and behavioral cloning to replayed experience to prevent the policy from drifting on past tasks, whereas our replay simply resamples earlier topologies into the same on-policy updates. The underlying design question — how much training budget to allocate to new versus past experience — is nonetheless the same, and in our setting, replay ratio is the dominant factor governing the generalisation–specialisation tradeoff. Warmup length, by contrast, has little effect on final performance in our setting at any topology size or replay ratio (Section 4.3). This suggests the relative weighting of historical versus current experience matters more when the goal is broad generalisation across a task distribution, as in curriculum learning, than when the goal is primarily preventing forgetting of previously mastered tasks, as in continual learning. Neither replay extreme dominates across all evaluation slices, and the factorial matrix experiment did not identify a single optimal configuration, suggesting the optimal strategy is likely task-dependent.

5.3 Revisiting the Sub-Questions

The preceding discussion bears directly on the paper's three sub-questions.

SQ1: both narrow-training baselines transfer measurably worse than any multi-size training condition, with the gap largest at sizes structurally distant from what was trained on. The partial recovery these baselines show at their own training size (q8) is a training-recency effect, as the heatmap demonstrates by isolating dosage from topology structure.

SQ2: curriculum variants outperform single-topology and single-size training on held-out topologies (Section 4.1). Against unordered exposure to the same topologies (random-curriculum), the advantage holds on frontier (q8) performance; on held-out generalisation, curriculum is indistinguishable from diversity alone.

SQ3: the no-replay and low-replay curricula reach higher q8 evaluation reward earlier in training than direct q8-only training, despite accumulating fewer q8-specific steps for the same total budget — consistent with transfer from smaller topologies improving sample efficiency.

Together, these findings support curriculum learning as a viable approach to topology-general qubit mapping, with the clearest benefits on frontier performance and sample efficiency, and a more qualified benefit — indistinguishable from simple topology diversity on this evaluation set — for generalisation to structurally novel hardware.

5.4 Limitations

Several limitations should be noted. All experiments are restricted to topologies of at most 8 qubits; whether the observed curriculum benefits persist at larger scales remains an open question. The held-out evaluation set contains only two topologies per size (Section 3.5), limiting the statistical power of generalisation claims independently of seed count; this is distinct from, and in addition to, the seed variance discussed below. Smaller training topologies are also structurally denser relative to their qubit count — a 4-qubit grid is close to fully connected — so randomly sampled interaction graphs are more likely to land on supported edges by chance at small sizes regardless of mapping quality; this may inflate reward at q4 across all conditions and should be considered when interpreting Figure 3. Interaction graphs are sampled randomly rather than drawn from real circuit benchmarks, so results may not reflect performance on practically relevant workloads. Noise and fidelity models are not considered: the reward signal optimises structural mapping quality but does not account for gate error rates or decoherence. Finally, several pairwise comparisons remain within one standard deviation of each other (Section 4.1), and the findings should be treated as indicative trends rather than definitive rankings.

6 Conclusions and Future Work

Quantum computers require a compilation step that maps abstract circuits onto the constraints of physical hardware. A critical part of this process is initial qubit mapping: assigning virtual qubits to physical qubits such that interacting pairs are placed on connected hardware locations. Poor mappings force the insertion of additional SWAP gates, which introduce noise and reduce the fidelity of the computation. This work investigated whether curriculum learning — training a reinforcement learning agent on hardware topologies of progressively increasing size — can produce an agent that generalises to unseen topologies, compared to training on a single fixed topology.

The central finding is that curriculum learning outperforms both single-topology and direct large-topology training on held-out and frontier metrics alike. All curriculum variants achieved higher performance than the q8-only and fixed-line8 baselines on held-out topologies; on q8, this held for the low-replay variants, while the high-replay 50/50 variant fell below both baselines on this metric specifically. The no-replay curriculum (100/0) also reached competitive 8-qubit performance faster, demonstrating improved sample efficiency. Against unordered exposure to the same topologies (random-curriculum), the benefit is narrower than it first appears: curriculum sharpens frontier (q8) specialisation, but does not measurably improve generalisation to structurally novel hardware beyond what topology diversity alone provides. This supports the hypothesis that representations learned on smaller topologies transfer to larger ones, providing a useful initialisation before the agent encounters harder tasks, while suggesting that the specific ordering of exposure matters more for frontier specialisation than for broad generalisation. The replay ratio — controlling how much training time is spent revisiting earlier topologies after advancing to a new one — introduces a clear, monotonic tradeoff between generalisation and specialisation. Higher replay (50/50) improves performance across the full evaluation distribution, including structurally different held-out topologies, but reduces performance on the largest frontier topologies. Lower replay (100/0) has the opposite effect. Warmup length, by contrast, had essentially no effect on final performance at any topology size or replay ratio, including under a no-warmup control. A factorial experiment over replay ratios and warmup lengths did not identify a single configuration that dominates across all evaluation slices, suggesting the optimal replay strategy is likely task-dependent.

These results establish curriculum learning as a promising direction for topology-general qubit mapping, but several open questions remain. All experiments were limited to topologies of at most 8 qubits; whether the benefits of curriculum learning persist at the scale of real devices (50–100+ qubits) is unknown. Interaction graphs were randomly generated rather than drawn from real circuit benchmarks, so practical performance may differ. Noise and hardware fidelity were not modelled. Future work should address these gaps by scaling to larger topologies, evaluating on standard circuit benchmarks such as MQT Bench [15], and incorporating hardware noise models into the reward signal. Motivated by the task-dependence of the optimal replay strategy observed above, adaptive curriculum scheduling offers an alternative to the fixed advancement thresholds and hand-chosen replay ratios used here, and may reduce sensitivity to hyperparameter choices. Candidates include ALP-GMM [13], which selects tasks by absolute learning progress; PAIRED [4], which generates a curriculum via regret-maximising environment design explicitly targeting zero-shot transfer; Teacher-Student Curriculum Learning [10]; and Prioritized Level Replay [7], which selects training tasks based on estimated learning progress. Hyperparameter tuning — in particular the learning rate, rollout length, and network architecture — was not performed in this work; systematic tuning may yield meaningful improvements in both sample efficiency and final performance. Finally, joint optimisation of mapping and routing remains an important open problem, as improvements in mapping quality do not always translate directly to reductions in final circuit depth.

7 Responsible Research

This work does not involve human subjects or personal data. Several broader considerations apply.

Use of AI tools. Large language models were used throughout this project as writing and coding assistants. Claude was used to help draft and revise prose in this report, including restructuring explanations for clarity and tightening discussion of results; OpenAI Codex was used to help write and debug training and evaluation code. Experimental design, the choice of curriculum strategies, the interpretation of results, and the analysis underlying the paper's claims are the author's own; LLM assistance was not used to generate research hypotheses or experimental findings. Literature search was also partly LLM-assisted, used to help locate and summarise candidate related work, which was then read and selected for inclusion by the author. All code and reported numbers were independently reviewed by the author before inclusion.

Computational cost. RL training across multiple seeds and configurations carries a non-trivial energy cost. The single-model approach investigated here is partly motivated by reducing the need to retrain a separate model per device topology, which would otherwise multiply this cost across hardware generations; if curriculum learning continues to scale favourably, it represents a path toward lower aggregate training cost than the per-topology status quo.

Reproducibility. All training and evaluation code, along with the configuration files needed to reproduce the experiments reported in this work, is available at https://github.com/alvov26/topology-curriculums.

Downstream implications. Advances in quantum compilation contribute, in principle, to making practical quantum computation more accessible, which has long-term implications for cryptographic security: sufficiently powerful quantum computers would break widely deployed public-key schemes such as RSA and ECC. This consideration is distant and indirect for the present work, which operates on toy topologies of at most 8 qubits, several orders of magnitude below the scale at which cryptographically relevant quantum computation becomes feasible.

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 41–48. ACM, 2009.

[2] Christopher Chamberland, Guanyu Zhu, Theodore J. Yoder, Jared B. Hertzberg, and Andrew W. Cross. Topological and subsystem codes on low-degree graphs with flag qubits. *Physical Review X*, 10:011022, 2020.

[3] Alexander Cowtan, Silas Dilkes, Ross Duncan, Alexandre Krajenbrink, Will Sheridan, and Seyon Sivarajah. On the qubit routing problem. *arXiv preprint arXiv:1902.08091*, 2019.

[4] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. In *Advances in Neural Information Processing Systems*, volume 33, pages 13049–13061, 2020.

[5] Chin-Yi Huang, Chen-Hung Lien, and Wai-Kei Mak. Reinforcement learning and DEAR framework for solving the qubit mapping problem. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–9. ACM/IEEE, 2022.

[6] Takehiro Ito et al. Algorithmic theory of qubit routing. In *Algorithms and Data Structures Symposium (WADS 2023)*, volume 14079 of *Lecture Notes in Computer Science*, pages 533–546. Springer, 2023.

[7] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4940–4950. PMLR, 2021.

[8] Gushu Li, Yufei Ding, and Yuan Xie. Tackling the qubit mapping problem for NISQ-era quantum devices. In *Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 1001–1014. ACM, 2019.

[9] Ying Li, Wenxuan Liu, and Ming Li. Deep reinforcement learning for mapping quantum circuits to 2D nearest-neighbor architectures. *Advanced Quantum Technologies*, 7(2):2300289, 2024.

[10] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher–student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740, 2019.

[11] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020.

[12] Rares Adrian Oancea et al. Optimizing initial qubit mappings under fixed gate error rates using deep reinforcement learning. In *Innovations for Community Services – 25th International Conference, I4CS 2025*, volume 2513 of *Communications in Computer and Information Science*, pages 189–208. Springer, 2025.

[13] Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep RL in continuously parameterized environments. In *Proceedings of the Conference on Robot Learning (CoRL)*, volume 100 of *Proceedings of Machine Learning Research*, pages 835–853, 2020.

[14] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

[15] Nils Quetschlich, Lukas Burgholzer, and Robert Wille. MQT Bench: Benchmarking software and design automation tools for quantum computing. *Quantum*, 7:1062, 2023.

[16] Antonin Raffin et al. Stable-baselines3 contrib, 2021.

[17] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[18] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[19] Marcos Yukio Siraichi, Vinícius Fernandes dos Santos, Sylvain Collange, and Fernando Magno Quintão Pereira. Qubit allocation. In *Proceedings of the 2018 International Symposium on Code Generation and Optimization (CGO)*, pages 113–125. ACM, 2018.

[20] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[21] Stan van der Linde et al. qgym: A gym for training and benchmarking RL-based quantum compilation. In *Proceedings of the 2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 2, pages 26–30. IEEE, 2023.

[22] Xiangzhen Zhou, Sanjiang Li, and Yuan Feng. Quantum circuit transformation based on simulated annealing and heuristic search. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(12):4683–4694, 2020.