Optimizing Collaborative Learning: A Hard-Constraint Reinforcement Learning Approach to Fair Team Composition

13th International Conference on Applied Informatics (ICAI 2026)

Marcell Herbák[1] Gergely Kovásznai[1] Ali Adil Adil[2]

[1]Eszterházy Károly Catholic University, Eger, Hungary
{herbak.marcell, kovasznai.gergely}@uni-eszterhazy.hu

[2]University of Debrecen, Debrecen, Hungary
ali.adil@unideb.hu

ICAI 2026 — pp. 133–145
DOI: 10.17048/icai.2026.133

Collaborative Learning in STEAM

Why Team Composition Matters

- Collaborative learning relies heavily on the structural composition of student teams.
- Well-balanced teams foster **peer-assisted learning**, active discussions, and higher overall engagement.
- When teams are imbalanced, pedagogical efficacy degrades significantly.

Consequences of Imbalanced Grouping

- **Demographic isolation:** Underrepresented students may be isolated in skewed groups.
- **Extreme skill variance:** High variance inside or across groups suppresses active participation and creates unfair advantages.

Homogeneous or Heterogeneous Groups?

Homogeneous grouping

"Overall, homogeneously grouped students (who were grouped based on achievement on the first test given in the course) significantly outperformed heterogeneously grouped students on the final examination."

(Baer, 2003, p. 169)

Heterogeneous grouping

"In mixed ability groups [...] the weaker students gain from seeing how better students study and approach problems, and the stronger students gain a deeper understanding of the subject by teaching it to others."

(Felder & Brent, 1994, p. 7)

Grouping Rules

- Educators must satisfy strict, simultaneous rules when forming student teams:
  1. Fixed team sizes (S students per group).
  2. Strict demographic bounds (e.g., maintaining a 40%–60% gender ratio).
  3. Balanced distribution of academic performance and engagement metrics.

- **Core Question:** How can we bridge the gap between exact mathematical optimality and real-time heuristic speed?

Class Size

Scaling to Real Classrooms

- University seminars and STEAM laboratory cohorts routinely contain $N = 40$ or more students, partitioned into K groups of size $S = 10$ (or $S = 5$).
- Every student carries a multi-dimensional profile:
  - Academic: *Exam Scores*, *Assignment Grades*
  - Behavioral: *Attendance*, *Discussions*, *Extracurriculars*, *EduTech Affinity*
  - Demographic: *Gender*

- Manually satisfying hard demographic bounds while balancing 7 distinct features across 40 students is administratively unscalable.

Bias in Manual Grouping

Student Self-Selection Bias

- Friends naturally group with friends.
- High-achieving students cluster together (**"super groups"**).
- Lower-performing or quieter students are left behind in **"failure clusters"**.
- Often leads to homogeneous groups and demographic segregation.

Instructor Cognitive Bias

- Educators rely on subjective impressions or a single salient metric (e.g., last exam grade).
- Impossible for a human to objectively weigh 7 continuous and discrete metrics simultaneously without algorithmic assistance.

Cost of Re-evaluating Groups

- **Cascade Effect of Swaps:** Moving a single student from Group A to Group B alters:
  - The gender ratios of both Group A and Group B.
  - The intra-group skill variance (standard deviation) of both groups.
  - The remaining unassigned student pool for subsequent groups.
- **Need for Dynamic Re-grouping:**
  - What happens if 2 students are absent on lab day, or a new student joins late?
  - Educators need **real-time, interactive software** (sub-second response) rather than overnight batch jobs.

Requirements for an Automated Tool

Four Pillars of Practical Team Formation

1. **Hard Constraint Guarantee:** Non-negotiable pedagogical rules (group size, gender bounds) must be respected without exception.
2. **Multi-Objective Quality:** Maximize student engagement and diversity while minimizing skill variance.
3. **Global Structural Fairness:** Eliminate "failure clusters" so the weakest team is still competitive and supported.
4. **Real-Time Execution:** Generate valid rosters in seconds (or milliseconds) for interactive classroom use.

Related Work: Heuristics and Exact Solvers

Metaheuristic Approaches (GA, PSO)

- **Examples:** Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) (Alberola et al., 2016).
- **Pros:** Computationally lightweight and fast.
- **Cons:** Frequently trapped in local optima, and struggle to guarantee strict adherence to hard constraints without fragile penalty tuning.

Exact Solvers (SMT, OMT, ILP)

- **Examples:** Z3 SMT/OMT solver (Adil et al., 2025; de Moura & Bjørner, 2008; Masina & Sebastiani, 2026).
- **Pros:** Formal mathematical guarantees of constraint satisfaction and local optimality.
- **Cons:** Traverse massive state spaces, leading to a severe runtime bottleneck as cohort size grows.

Related Work: Machine Learning for Solvers and Safe RL

- **Machine Learning for Logic Solvers:**
  - Neural networks have been used to classify SAT problem instances and configure solvers to reduce runtime (Danisovszky et al., 2020).
  - However, hybrid classifiers still rely on the underlying exact solver's combinatorial search architecture.
- **Reinforcement Learning & Safe RL:**
  - Deep Q-Networks (DQN) excel at high-dimensional sequential decisions (Mnih et al., 2015).
  - Standard RL maximizes cumulative reward without respecting strict safety boundaries.
  - In autonomous driving, **Long and Short-Term Constraints (LSTC)** enforce strict safety bounds (García & Fernández, 2015; Hu et al., 2024).
  - **Our contribution:** Adapt LSTC Safe RL and Deep Q-learning from Demonstrations (Hester et al., 2018) to educational team formation.

Computational Complexity

Why is Constrained Team Composition NP-Hard?

- Partitioning a set of N weighted items into K equally sized subsets while satisfying multiple linear capacity/ratio bounds and minimizing intra-group variance generalizes classic NP-hard problems (e.g., *Multi-Dimensional Knapsack* and *Balanced Graph Partitioning*).
- No known polynomial-time algorithm can compute the exact global optimum for arbitrary student feature distributions.

- Adding both **hard linear bounds** (demographic ratios) and **non-linear dispersion objectives** (skill variance across groups) couples every student assignment decision with all others.

Size of the Search Space

Counting the State Space for $N = 40, S = 10, K = 4$

The number of ways to partition $N = 40$ students into $K = 4$ groups of size $S = 10$ is given by the multinomial coefficient (divided by $K!$ for unlabeled groups):

$$\frac{N!}{(S!)^K \cdot K!} = \frac{40!}{(10!)^4 \cdot 4!} \approx 1.96 \times 10^{20} \text{ distinct partitions}$$

- Even for choosing **just a single group** of $S = 10$ out of $N = 40$ students:

$$\binom{40}{10} = 847{,}660{,}528 \text{ candidate combinations}$$

- As classroom size N increases linearly, the search space grows factorially, creating an insurmountable wall for exhaustive exact evaluation.

SMT and QF_LIA

Satisfiability Modulo Theories (SMT)

- **Decision Problem:** Determines whether a Boolean formula is satisfiable with respect to a formal background theory (Barrett et al., 2025).
- Common theories include integers, real numbers, arrays, and fixed-size bit-vectors.

Our Fragment: QF_LIA

- We restrict our formalization to the **Quantifier-Free Linear Integer Arithmetic (QF_LIA)** fragment within the SMT-LIB standard.
- Standard SMT answers a binary question: *Does a valid team assignment exist?* (SAT / UNSAT).
- However, SAT alone does not guarantee the **best** pedagogical team.

Optimization Modulo Theories

From Satisfiability (SMT) to Optimization (OMT)

- **Optimization Modulo Theories (OMT)** extends SMT by incorporating formal objective functions to be maximized or minimized:

$$\max \text{obj} \quad \text{or} \quad \min \text{obj}$$

- The solver searches across all satisfying assignments of the `QF_LIA` constraints to find the assignment that optimizes obj.

- **Cost of OMT:** To prove that an assignment is the true maximum, the OMT solver (such as Microsoft's **Z3**) must exhaustively prune or verify that no strictly better satisfying assignment exists in the search tree.

Formalization: Decision Variables

Cohort and Group Definitions

- Let $U = \{u_1, u_2, \ldots, u_N\}$ be the finite set of N students.
- We want to partition U into K groups, each of exact target size S (where $N = K \cdot S$).

Binary Decision Variables

For every student $u_i \in U$ ($i \in \{1, \ldots, N\}$) and every group $k \in \{1, \ldots, K\}$, we define a binary decision variable $x_{i,k} \in \{0, 1\}$:

$$x_{i,k} = \begin{cases} 1 & \text{if student } u_i \text{ is assigned to group } k \\ 0 & \text{otherwise} \end{cases}$$

Formalization: Size and Assignment Constraints

1. Group Size Constraints

Each group k must contain exactly S students:

$$\sum_{i=1}^{N} x_{i,k} = S \quad \forall k \in \{1, \ldots, K\}$$

2. Student Assignment Constraints

Every student u_i must be assigned to **exactly one** group (no student left behind, no student duplicated):

$$\sum_{k=1}^{K} x_{i,k} = 1 \quad \forall i \in \{1, \ldots, N\}$$

Formalization: Gender Constraints

3. Bounded Gender Ratio per Group

Let $g_i \in \{0, 1\}$ represent the binary demographic indicator (gender) of student u_i. The proportion of students with $g_i = 1$ in any group k must lie within the pedagogical lower bound α and upper bound β:

$$\alpha \cdot S \leq \sum_{i=1}^{N} x_{i,k} \cdot g_i \leq \beta \cdot S \quad \forall k \in \{1, \ldots, K\}$$

- **In our experimental setup:**
  - Target group size: $S = 10$
  - Demographic bounds: $\alpha = 0.4$, $\beta = 0.6$
  - Every 10-person team must contain between **4 and 6** students of the specified gender.

Formalization: Composite Score

Weighted Composite Feature Score (CS_i)

Each student u_i possesses M normalized academic and engagement features denoted by $f_i^1, \ldots, f_i^M$. We define the student's composite score CS_i as:

$$CS_i = \sum_{j=1}^{M} w_j \cdot f_i^j \tag{3.1}$$

where w_j is the predefined pedagogical weight for the j^{th} feature.

Feature	Exam Score	Assignment	Attendance	Discussions	Extracurr. / EduTech
Weight (w_j)	0.7	0.3	0.2	0.1	0.1 each

Baseline: Iterative OMT

Global Intractability

- Computing a single, simultaneous global optimum across all K groups with non-linear variance minimization is computationally intractable for Z3 at $N = 40$.

Iterative & Greedy OMT Baseline Design

- To make the exact solver runnable, we decompose the problem into K sequential OMT calls.
- Instead of global variance minimization, each OMT call maximizes the composite score for the **current group** k **being formed** from the remaining unassigned pool $U_{\text{unassigned}} \subseteq U$.

Baseline: One Iteration

The k^{th} Iteration Optimization Problem

At step $k \in \{1, \ldots, K\}$, the Z3 OMT solver evaluates the remaining unassigned students $U_{\text{unassigned}} \subseteq U$ and solves:

$$\max \sum_{u_i \in U_{\text{unassigned}}} x_{i,k} \cdot CS_i \tag{3.2}$$

subject to size ($\sum x_{i,k} = S$) and demographic ($\alpha S \leq \sum x_{i,k} g_i \leq \beta S$) constraints.

1. Solve Equation (3.2) to optimality using Z3 for group k.
2. **Freeze** the returned assignments $\{u_i \mid x_{i,k} = 1\}$.
3. **Remove** those S students from $U_{\text{unassigned}}$ and proceed to group $k + 1$.

Dataset

- **Dataset Source:** Statistically accurate cohort of $N = 40$ students based on standard higher-education distribution curves, adapted from Najem (2025).
- **7 Intrinsic Student Features:**
  1. Exam Scores (continuous, normalized)
  2. Assignment Grades (continuous, normalized)
  3. Attendance Rate
  4. Discussion Participation
  5. Extracurricular Involvement
  6. EduTech Affinity
  7. Gender (binary indicator $g_i \in \{0, 1\}$)
- **Open Science:** Full dataset, Z3 baseline, and DRL code publicly available on GitHub.

Baseline Results: Solution Quality

Strengths of the Z3 OMT Baseline

- **100% Constraint Satisfaction:** Every formed group strictly obeys $S = 10$ and $\alpha = 0.4 \leq \text{ratio} \leq 0.6 = \beta$.
- **High Average Engagement:** Achieves an average engagement score of **1.201** and diversity count of **12.5**.
- **Low Average Skill Variance:** Achieves an intra-group skill variance of **0.198**.

- On paper, the *average* metrics across the classroom look mathematically strong.
- However, looking behind the averages reveals two severe operational flaws.

Baseline Results: Runtime

Severe Computational Cost

- Even though the problem was simplified into iterative greedy steps, Z3 required **13,304 seconds** (**≈ 3.7 hours**) to compute the distribution for just $N = 40$ students!

- Why does it take hours?
  - In the early iterations ($N = 40$, then $N = 30$), Z3 must prove optimality over hundreds of millions of valid subset combinations with real-valued objective sums.
  - Completely precludes real-time or interactive classroom software.

Baseline Results: Groups by Iteration

Iteration / Group	Remaining Pool Size	OMT Behavior	Team Quality
Group 1 ($k = 1$)	$40 \rightarrow 30$	Skims top performers	Elite "Super Group"
Group 2 ($k = 2$)	$30 \rightarrow 20$	Skims next best available	Above Average
Group 3 ($k = 3$)	$20 \rightarrow 10$	Limited high-performers left	Below Average
Group 4 ($k = 4$)	$10 \rightarrow 0$	Forced to take leftovers	"Failure Cluster"

- Resulting **Group Score Gap** (difference between strongest and weakest team): **0.174**.

Greedy Deterioration

Definition: The Greedy Deterioration Problem

- Because the iterative OMT solver maximizes Equation (3.2) at each individual step, it **aggressively groups the highest-performing students together** in the early iterations.
- By the final iteration, the pool of unassigned students $U_{\text{unassigned}}$ is completely depleted of high-performers.
- This deterministic "skimming" guarantees the creation of a **"failure cluster"** at the end of execution.

- Local optimality at step $k = 1$ directly causes global unfairness at step $k = K$.

Limitations of Iterative OMT

Limitations of Iterative OMT

1. **Myopic (Short-Sighted):** Cannot foresee how forming a "super group" now hurts future groups.
2. **Intractable Runtime:** 13,304 s (3.7 hours) for $N = 40$.
3. **Zero Reusability:** If one student changes, the entire 3.7-hour solve must be repeated from scratch.

How RL Solves This

1. **Delayed Gratification:** RL maximizes long-term cumulative return, proactively reserving strong students for later teams.
2. **Linear-Time Inference:** Once trained, generating a roster takes < 0.04 seconds.

MDPs and Deep Q-Networks

Markov Decision Process (MDP) (Sutton & Barto, 2018)

Sequential decision-making tuple $(\mathcal{S}, \mathcal{A}, P, R)$:

- At discrete time step t, the agent observes state $s_t \in \mathcal{S}$ and selects action $a_t \in \mathcal{A}$.
- The environment transitions to $s_{t+1} \in \mathcal{S}$ and emits immediate reward r_t.
- **Objective:** Learn a policy $\pi(a \mid s)$ that maximizes expected cumulative return.

Deep Q-Networks (DQN) (Mnih et al., 2015)

- Approximates the optimal action-value function $Q^*(s_t, a_t)$ using a deep neural network (MLP).
- Trained by minimizing Temporal Difference (TD) error between predicted and target Q-values using an experience replay buffer $\mathcal{D}$ and soft target updates ($\tau = 0.01$).

Safe RL with Long- and Short-Term Constraints

Why Standard DRL Fails on Hard Constraints

- Standard DRL agents happily violate demographic rules if doing so yields a high engagement bonus, or spend thousands of episodes exploring invalid states.

Adapting LSTC Architecture (Hu et al., 2024) to EdTech

We separate constraints into two distinct horizons:

1. **Short-Term (Immediate Step-Level) Hard Constraints:** Enforced instantaneously at every candidate evaluation via **Dynamic Action Masking** and **Safety Overrides**.
2. **Long-Term (Episode/Terminal-Level) Constraints:** Enforced across the formed roster via **Cubic Reward Shaping** ("Reward Cliff") and skill variance penalties.

Team Formation as an MDP

- Instead of searching over 1.96×10^{20} global partitions, we frame team formation as a **sequential candidate evaluation process**:

  1. The environment presents one unassigned candidate student u_i at a time to the agent.
  2. The agent inspects the student's intrinsic profile **alongside** the real-time context of the currently forming group and remaining student pool.
  3. The agent decides whether to **Accept** ($a_t = 1$) or **Reject** ($a_t = 0$) the candidate into the current group.
  4. If rejected, the student remains in the unassigned pool for later evaluation. Once a group reaches size S, a terminal episode reward r_{terminal} is computed.

State Space: Student Features

Category 1: Intrinsic Student Features (7 Dimensions)

At step t, the first part of the observation vector $s_t \in \mathcal{S}$ encodes the normalized dataset features of the current candidate student being evaluated:

1. **Exam Score** $\in [0, 1]$
2. **Assignment Grade** $\in [0, 1]$
3. **Attendance Rate** $\in [0, 1]$
4. **Discussion Participation** $\in [0, 1]$
5. **Extracurricular Involvement** $\in [0, 1]$
6. **EduTech Affinity** $\in [0, 1]$
7. **Gender** $g_i \in \{0, 1\}$

State Space: Context Variables

Category 2: Real-Time Episode Context (4 Dimensions)

To make non-greedy, globally aware decisions, the agent also observes real-time summary statistics of the environment:

1. **Normalized Current Group Size:** How full the current group is relative to target size S.
2. **Current Group Male Ratio:** Real-time demographic balance of the team being built.
3. **Unassigned Pool Male Ratio:** Demographic availability remaining in $U_{\text{unassigned}}$ (crucial for anticipating future bottlenecks!).
4. **Running Average Exam Score:** Current academic mean of the partially formed group.

- Combining Intrinsic + Context features gives the MLP policy full visibility into both the individual candidate and the global trajectory.

Action Space

Binary Discrete Action Space $\mathcal{A} = \{0, 1\}$

- $a_t = 1$ (**Accept**): Add the candidate student to the current group.
- $a_t = 0$ (**Reject**): Pass on the candidate for now (keep in unassigned pool).

Environment as Strict Gatekeeper

- The agent's raw output a_t is never executed blindly.
- Before mutating the state, the environment checks whether a_t violates short-term hard constraints or risks a pool deadlock.

Reward Function

Composite Terminal Reward Formulation

When a group reaches target size S, the environment computes the terminal reward:

$$r_{\text{terminal}} = r_{\text{base}} - P_{\text{gender}} - \sigma \cdot P_{\text{skill}} + B_{\text{engage}} + B_{\text{diverse}} \tag{3.4}$$

- $r_{\text{base}} = 100$: Base completion reward.
- $P_{\text{gender}} = \gamma \cdot |\Delta\text{ratio}|^3$ ($\gamma = 2$): Cubic penalty for demographic deviation.
- P_{skill} ($\sigma = 100$): Weighted standard deviation of the group's Exam Scores (0.7) and Assignment Grades (0.3).
- B_{engage}: Bonus weighting group mean Attendance (5.0) and Discussions (5.0).
- B_{diverse}: Bonus weighting aggregated EduTech Affinity (5.0) and Extracurriculars (3.0).

Replay Buffer Pre-loading (DQfD)

Algorithm 1: DQfD Initialization

1. Initialize replay buffer $\mathcal{D}$ and environment E.
2. **for** step $= 1$ to E_{steps} **do**
3. Observe state s_t, evaluate bounds α, β.
4. **if** Acceptance violates bounds **or**
5. (avg_score $> 0.8 \wedge$ cand > 0.8) **or**
6. (avg_score $< 0.65 \wedge$ cand < 0.65) **then**
7. $a_t \leftarrow 0$ *(Expert rejection)*
8. **else** $a_t \leftarrow 1$ *(Expert acceptance)*
9. Execute a_t, store (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$.
10. Begin DQN training with pre-loaded $\mathcal{D}$.

Why Expert Demonstrations?

- Standard DRL starts completely blind.
- Our heuristic teacher actively rejects high-performers (> 0.8) if group mean > 0.8 (preventing **super groups**) and rejects low-performers if group mean < 0.65 (preventing **failure clusters**).
- Anchors gradient descent to a valid baseline from step 1.

Constraint Handling: Action Masking

Layer 1: Dynamic Action Masking (Short-Term Gatekeeper)

- Suppose the policy outputs $a_t = 1$ (Accept), but adding this candidate would push the group's male or female count beyond the hard bound $[\alpha \cdot S, \beta \cdot S]$.
- The environment intercepts the invalid transition before it occurs:

$$\texttt{valid_add} \leftarrow \texttt{False}, \quad a_t \text{ forced to 0 (Reject)}$$

- Simultaneously, the environment applies an **immediate localized penalty of** -10 to discourage the neural network from proposing invalid branches.

Constraint Handling: Safety Overrides

Layer 2: Preventing Infinite Rejection Deadlocks

- Because the agent is allowed to reject candidates ($a_t = 0$) in search of better combinations, a naive policy might reject too many students and run out of candidates to fill the remaining slots.
- **Deterministic Force-Fill Override:**

$$\text{If } |U_{\text{remaining}}| = S - |G_{\text{current}}|, \quad \text{force } a_t \leftarrow 1$$

- When the number of remaining students in the pool exactly equals the number of empty slots needed to complete the group, the environment bypasses the policy and force-fills the roster, guaranteeing **zero deadlocks**.

Constraint Handling: Cubic Penalty

The "Reward Cliff": Polynomial Taxation

Linear penalties are often overpowered by secondary engagement bonuses. We introduce a cubic penalty for demographic deviation:

$$P_{\text{gender}} = \gamma \cdot |\Delta\text{ratio}|^3 \tag{3.3}$$

- Small deviations near the ideal ratio receive negligible taxation, allowing smooth gradient flow for skill and engagement optimization.
- Larger deviations hit a steep **"Reward Cliff"** where $|\Delta\text{ratio}|^3$ explodes, mathematically obliterating accumulated positive bonuses.
- Trained in Stable-Baselines3 (Raffin et al., 2021) for 50,000 timesteps ($\text{lr} = 10^{-3}$).

Inference Procedure

How the trained agent forms the groups

- Groups are formed one at a time from the unassigned pool U_k, starting with $U_1 = U$.
- For group k, the agent is shown a window $C_k \subseteq U_k$ of $|C_k| = \min(W, |U_k|)$ candidates, $W = \lfloor 1.5\,S \rfloor = 15$, one by one.
- The accepted students form G_k and leave the pool: $U_{k+1} = U_k \setminus G_k$.

Two counters and two automatic rules (at step t)

- r_t = candidates left in the window, and $n_t = S - |G_t|$ = empty seats (group size minus students accepted so far).
- **Force-fill:** if $r_t \leq n_t$, the candidate is accepted regardless of the agent.
- **Masking:** an acceptance that would put more than $\lfloor \beta S \rfloor = 6$ students of one gender into the group is turned into a rejection.

Deadlock-Freedom

Every group gets filled, whatever the agent decides

If at least S students are unassigned ($|U_k| \geq S$), every group ends with exactly $|G_k| = S$ students.

Why this holds: there are never fewer candidates than empty seats, $r_t \geq n_t$

- At the start: $r_0 = \min(15, |U_k|) \geq 10 = n_0$.
- If $r_t > n_t$: one candidate is used up and at most one seat is filled, so $r_{t+1} = r_t - 1 \geq n_t \geq n_{t+1}$.
- If $r_t = n_t$: force-fill accepts, so both drop together, $r_{t+1} = r_t - 1 = n_t - 1 = n_{t+1}$.
- So when the window runs out ($r_t = 0$), also $n_t = 0$: the group is full.

The argument never uses the agent's decisions, which matches the 0 deadlocks in all 50 ablation runs.

From Episodes to the Formal Model

As in the formal model, let $x_{i,k} = 1$ if student u_i ends up in group G_k, and $x_{i,k} = 0$ otherwise.

The agent's output is a valid solution of the formal model (for $N = K \cdot S$)

$$\sum_{i=1}^{N} x_{i,k} = S \quad \forall k \qquad \text{and} \qquad \sum_{k=1}^{K} x_{i,k} = 1 \quad \forall i.$$

Why this holds

Before group k, $|U_k| = N - (k-1)S \geq S$ students are left, so every group is filled (Slide 39). Assigned students leave the pool, so nobody is chosen twice, and K groups of S cover all $N = KS$ students.

Linear effort

The number of agent decisions is $\sum_k |C_k| \leq K \cdot W \leq 1.5\,N$, i.e. at most $3 \cdot 15 + 10 = 55$ for $N = 40$. The work grows linearly with the class size.

Gender Bounds Under Action Masking

Groups built by the agent's own decisions respect the gender bounds

If no student of a group was force-filled, its number of male students m satisfies

$$\alpha S \le m \le \beta S, \qquad \text{i.e. } 4 \le m \le 6 \text{ for } S = 10.$$

Why this holds

Masking keeps $m \le \beta S$ and $f \le \beta S$ at all times. A full group has $m + f = S$, so $m = S - f \ge (1 - \beta)S = \alpha S$, because $\alpha + \beta = 1$.

Limitation

- Force-filled students bypass the mask. The last group is always exactly the S students left over ($r_0 = n_0 = S$), so its balance depends on the earlier groups.
- This is why the agent also sees the gender ratio of the remaining pool.

Penalized Proposals Are Strictly Suboptimal

$Q^*(s, a)$ is the best total reward the agent can still collect after action a in situation s ($a = 1$ accept, $a = 0$ reject), and $V^*(s')$ is the best reward collectable from the next situation s'.

Proposing a blocked student is always exactly 10 points worse

$$Q^*(s, 1) = Q^*(s, 0) - 10.$$

Why this holds

A blocked acceptance leaves the group unchanged, exactly like a rejection, so both lead to the same next situation s'. Only the immediate reward differs (-10 versus 0). With the discount factor λ:

$$Q^*(s, 1) = -10 + \lambda\, \mathbb{E}[V^*(s')] = Q^*(s, 0) - 10.$$

A trained agent therefore stops proposing blocked acceptances. The expert demonstrations (DQfD) also reject in these situations.

Size of the Cubic Penalty

Let $d = |m - S/2|$ be how many students a group is away from a 5/5 split. With $\gamma = 2$, the gender penalty is $P_{\text{gender}} = 2d^3$.

Upper bound on the reward

Each bonus is a weight times a value in $[0, 1]$, so $0 \le B_{\text{engage}} + B_{\text{diverse}} \le 5 + 5 + 5 + 3 = 18$. Since $P_{\text{skill}} \ge 0$,

$$r_{\text{terminal}} \le 100 + 18 - 2d^3.$$

d	0	1	2	3	4	5
Male students m ($S = 10$)	5	4 or 6	3 or 7	2 or 8	1 or 9	0 or 10
Within $[\alpha S, \beta S]$	yes	yes	no	no	no	no
Cubic penalty $2d^3$	0	2	16	54	128	250
Linear penalty $2d$ (for comparison)	0	2	4	6	8	10

- Inside the bounds the penalty is at most 2. From $d = 3$ it is larger than all bonuses together ($54 > 18$), and from $d = 4$ the whole reward is negative.
- A linear penalty with the same γ never outweighs the bonuses ($2d \le 10 < 18$).

Why Iterative OMT Front-Loads Strong Groups

Let $\Phi(G) = \sum_{u_i \in G} CS_i$ be the total composite score of group G, the quantity OMT maximizes in each iteration (3.2).

OMT's groups get weaker with every iteration

$$\Phi(G_1) \geq \Phi(G_2) \geq \cdots \geq \Phi(G_K).$$

Why this holds

$G_{k+1} \subseteq U_{k+1} \subseteq U_k$, so its students were still available when OMT chose G_k, and G_{k+1} also satisfies all constraints. OMT picked G_k as the best option, so $\Phi(G_k) \geq \Phi(G_{k+1})$.

Consequence

All groups have size S, so with the class average $\overline{CS}$ we get $\Phi(G_1)/S \geq \overline{CS} \geq \Phi(G_K)/S$. The first group is above average and the last below it (iteration table, Slide 24). The failure cluster is built into the method.

What the RL Reward Does and Does Not Reward

The reward looks at the spread of the scores, not their level

Exam and assignment scores enter r_{terminal} only through their standard deviation in P_{skill}, and for any constant c

$$\text{sd}(x_1 + c, \ldots, x_S + c) = \text{sd}(x_1, \ldots, x_S).$$

Raising every member's score by the same amount does not change the reward.

- OMT's objective Φ grows with every exam score ($w = 0.7 > 0$), but the RL reward does not. So the argument of Slide 44 does not apply to the agent, which gains nothing by collecting the strongest students into one group.

Rejecting is cheap

A rejection earns no immediate reward, so $Q^*(s, 0) = \lambda\, \mathbb{E}[V^*(s')]$. Waiting n steps only scales the final reward by λ^n. With the Stable-Baselines3 default $\lambda = 0.99$, $\lambda^{50} \approx 0.61$, which allows the long, selective episodes in Figure 1.

A Direct Translation of the OMT Model

What if the reward simply copied OMT? The group would get the OMT objective (3.2), and a gender violation would cost a large penalty M:

$$r_{\text{dir}}(G) = \Phi(G) - M \cdot \mathbf{1}\big[\, m \notin [\alpha S, \beta S] \,\big].$$

Copying OMT reproduces OMT

If the agent sees the whole pool ($W \geq |U_k|$) and does not discount ($\lambda = 1$), the best possible policy picks exactly OMT's group:

$$V^*(s_0) = \max_{T \subseteq U_k \text{ valid}} \Phi(T).$$

Why this holds

The agent can form any valid group T by accepting exactly its members, and invalid groups score lower because of M. So the best achievable reward is the best valid Φ, which is OMT's choice.

Group by group, the agent would repeat OMT, so Slide 44 applies: the same failure cluster, only faster.

Requirements on the Gender Penalty

Let $P(d)$ be the gender penalty. Features lie in $[0, 1]$, so a standard deviation is at most $\frac{1}{2}$ and $0 \leq \sigma P_{\text{skill}} \leq 50$. The bonuses lie in $[0, 18]$.

- **(T) Tolerance:** a one-student slip ($5/5 \to 4/6$) costs less than the bonuses, $P(1) - P(0) < 18$.
- **(D) Dominance at d:** a group with deviation d scores below every acceptable group,

$$\underbrace{118 - P(d)}_{\text{best group with deviation } d} < \underbrace{50 - P(1)}_{\text{worst acceptable group}} \iff P(d) - P(1) > 68.$$

No linear penalty $P(d) = c \cdot d$ meets both at $d = 4$

(T) needs $c < 18$. (D) at $d = 4$ needs $3c > 68$, i.e. $c > 22.6$. Both cannot hold.

Choosing the Exponent

Penalty $P(d)$	**(T):** $P(1) - P(0)$	**(D) holds from**	**Increasing in** d
Indicator $M \cdot \mathbf{1}[d \geq 2]$, $M > 68$	0	$d = 2$	no
Linear $2d$	2	never (max $= 8$)	yes
Quadratic $2d^2$	2	never (max $= 48$)	yes
Cubic $2d^3$	**2**	$\mathbf{d = 4}$ ($126 > 68$)	**yes**

- The indicator (a flat fine) meets (T) and (D) but treats a 3/7 split the same as 0/10, and does not prefer 5/5 over 4/6.
- With $\gamma = 2$, the cubic is the simplest power that is increasing, tolerant, and dominant from $d = 4$. Splits with $d = 2, 3$ are already blocked by masking (Slide 41).
- Each extra student of imbalance costs more than the previous one,

$$P(d) - P(d-1) = 2(3d^2 - 3d + 1) = 2,\ 14,\ 38,\ 74,\ 122,$$

whereas a linear penalty charges the same every time.

Runtime and Solution Quality

Computational Efficiency ($N = 40$)

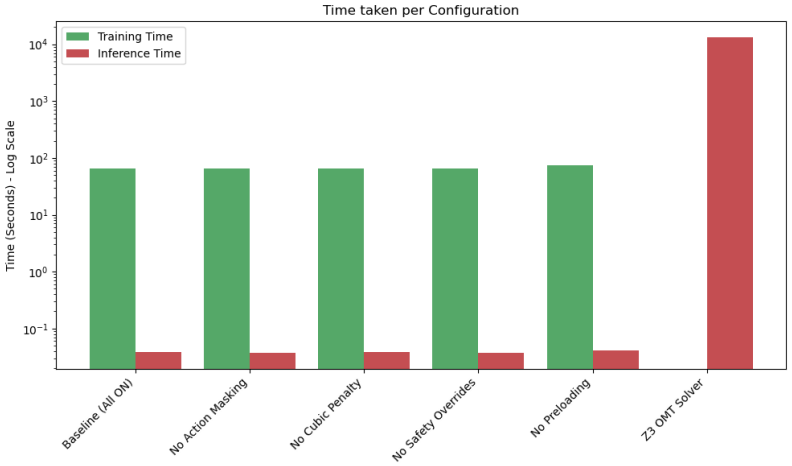
- **Iterative OMT (Z3):** 13,304 s ($\approx$ 3.7 hours)
- **DRL End-to-End (Train + Infer):** $\approx$ 65 s (**204$\times$ speedup**)
- **DRL Zero-Shot Inference:** $<$ 0.04 s (0.038 s) — linear time $\mathcal{O}(N)$!

Table: Performance Comparison ($N = 40$)

Metric	OMT (Z3)	DRL Agent
Avg. Engagement	1.201	**1.201**
Diversity Count	12.5	**12.5**
Skill Variance	**0.198**	0.200
Group Score Gap	0.174	**0.140**

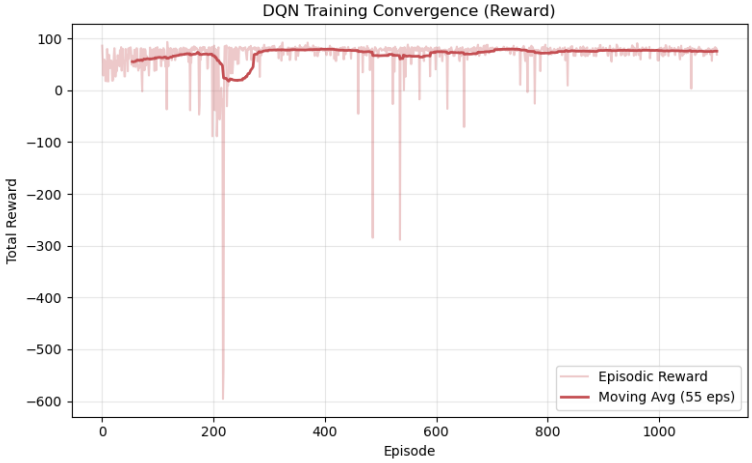
- **Key Finding:** DRL matches exact OMT on engagement and diversity while reducing the worst-case **Group Score Gap from** 0.174 **to** 0.140 **(**20% **fairness improvement)**, completely eliminating the "failure cluster"!

Runtime per Configuration



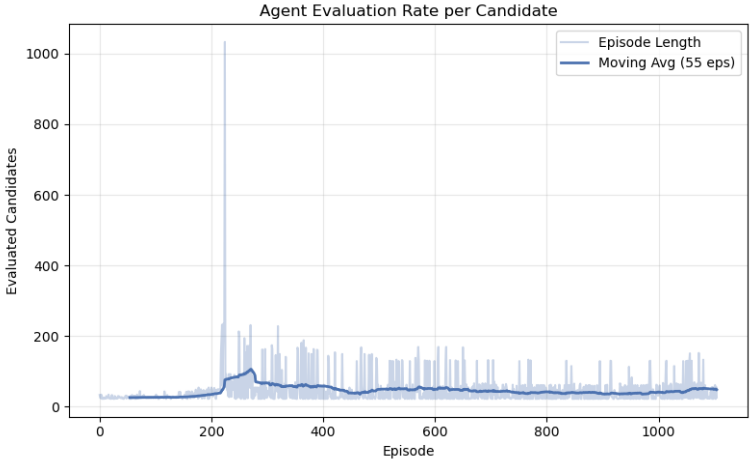
Training and inference time of each DRL ablation configuration compared with the Z3 OMT solver.

Training Reward



Episodic reward during DQN training, with moving average.

Candidates Evaluated per Episode



Agent evaluation rate per candidate, demonstrating non-greedy behavior.

Agent Behavior During Training

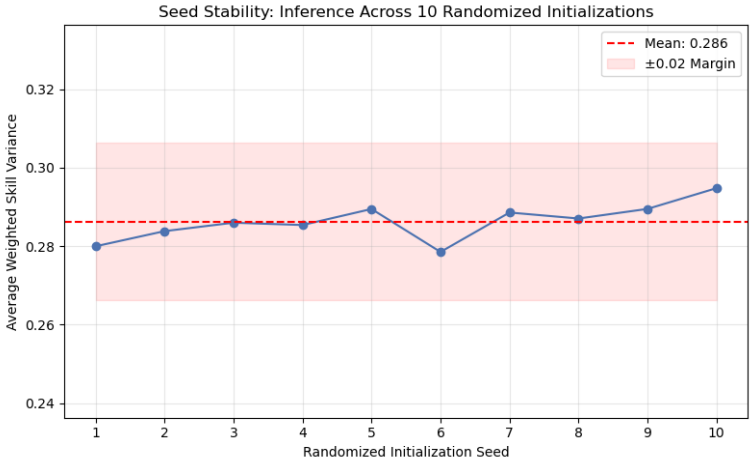
Figure 1 Analysis: Students Viewed per Group Formed

- In our behavioral analysis over 1,000 episodes (forming 5-person rosters where minimum possible steps $= 5$):
  - Early in training (Episodes 0–200), the agent views $\approx$ 25–30 candidates per group.
  - Around Episode 220, the policy transitions to a **"picky" regime**, evaluating 50 **to** 70 **candidate steps** per group formed (50-episode moving average stabilizes around 55–65 steps).

Proof of Delayed Gratification

- Why view up to 70 candidates to fill 5 slots? The agent **proactively rejects high-performing students early in the episode** if accepting them would skew future demographic bounds or create an unbalanced cohort later.

Seed Stability



Seed stability analysis over 10 randomized initialization states.

Seed Stability: Values

Figure 2 Analysis: Inference Across 10 Randomized Initializations

- A common concern with neural heuristics is sensitivity to input ordering. We tested the trained baseline policy across **10 fully randomized cohort initialization seeds**.

Seed	1	2	3	4	5	6	7	8	9	10
Skill Var.	0.280	0.284	0.286	0.285	0.289	0.279	0.289	0.287	0.289	0.295

- **Mean Weighted Skill Variance:** 0.286
- **Maximum Deviation from Mean:** < 0.009 (strictly bounded within a narrow ± 0.01 margin, well inside the ± 0.02 tolerance band).
- Proves that team quality stems from learned neural weights, not lucky initial student ordering.

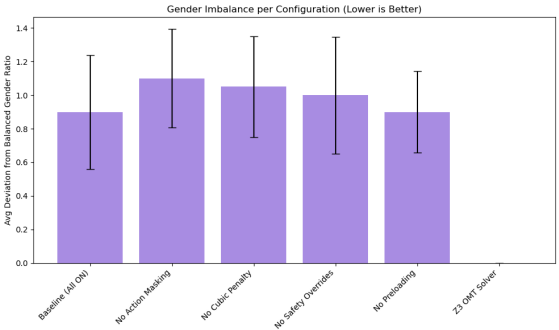
Ablation Study

Table: Ablation Study Results (Mean $\pm$ 95% CI over $N = 10$ seeds)

Configuration	Skill Variance	Gender Imbalance	Train Time (s)	Inference Time (s)
1. Baseline (All ON)	0.205 ± 0.007	$\mathbf{0.90 \pm 0.33}$	65.09 ± 1.52	0.038 ± 0.005
2. No Action Masking	0.203 ± 0.008	1.10 ± 0.29	65.41 ± 1.83	0.037 ± 0.002
3. No Cubic Penalty	0.208 ± 0.008	1.05 ± 0.30	64.53 ± 1.39	0.038 ± 0.003
4. No Safety Overrides	0.198 ± 0.007	1.00 ± 0.34	65.42 ± 2.20	0.037 ± 0.007
5. No Preloading (DQfD)	0.209 ± 0.009	0.90 ± 0.24	73.60 ± 6.91	0.040 ± 0.003

- **Zero Deadlocks:** Across all 50 executions (5 configs $\times$ 10 seeds), **0 deadlocks occurred**.
- Disabling **Action Masking** or **Cubic Penalty** spikes gender imbalance ($0.90 \rightarrow 1.10/1.05$).
- Disabling **DQfD Pre-loading** degrades training convergence time (65.09 s $\rightarrow$ 73.60 s) and skill variance.

Ablation Study: Charts





Gender imbalance (left) and weighted skill variance (right) per configuration vs. Z3 OMT (lower is better).

Summary of Results

Key Takeaways from Section 4

1. **Combinatorial Bottleneck Resolved:** Reduced total runtime from 3.7 hours (Z3 OMT) to 65 seconds (training + inference), and < 0.04 seconds for zero-shot inference on new cohorts.
2. **Elimination of Failure Clusters:** Improved worst-case inter-group fairness by **20%** (Group Score Gap reduced from 0.174 to 0.140) via delayed gratification.
3. **Verified Determinism & Safety:** Zero state deadlocks across 50 ablation runs and < 0.009 maximum variance deviation across 10 randomized seeds.

Feedback from the First Presentation

Points raised

1. The ablation study and the seed-stability test each used **only 10 runs**.
2. All experiments ran on a **single, fixed machine setting**.
3. **No different classroom settings** were tested, only one cohort configuration ($N = 40$, $S = 10$, $\alpha = 0.4$, $\beta = 0.6$).

The most debated point

- Is the gender ratio relevant to education? In other words, is a gender bound a meaningful pedagogical criterion for team composition, or only a constraint the method is able to enforce?

Future Work from the Paper

1. Scalability to Massive & Diverse Cohorts

- Evaluate zero-shot inference and training scaling on significantly larger university cohorts ($N = 200$ to $N = 1{,}000+$ students) and diverse demographic distributions.

2. Solver Completeness & Tight UNSAT Regimes

- While our agent achieved 0 deadlocks empirically, formal **completeness** remains open: how does the agent behave when demographic constraints are so tight that the instance approaches `UNSAT`?

3. Side-by-Side Comparison with Global ILP Optimizers

- Benchmark our Safe DRL framework directly against commercial **Integer Linear Programming (ILP)** formulations to map the exact Pareto frontier between heuristic speed and global mathematical optimality.

Future Work: Architectures

Modern RL & Graph Architectures Beyond DQN

- **Maskable PPO (Proximal Policy Optimization):**
  - Instead of penalizing invalid actions in a value-based DQN, apply **logit-level action masking** directly inside an Actor-Critic policy gradient framework (invalid actions receive $-\infty$ logit before softmax).
- **Attention / Set-Transformer & Graph Neural Network (GNN) Encoders:**
  - Currently, the MLP inspects one candidate at a time plus summary statistics.
  - A **Set-Transformer** or **GNN** state encoder can attend over the *entire unassigned student pool simultaneously*, making the policy completely permutation-equivariant and invariant to cohort size N.

Future Work: Hybrid Solvers and Peer Constraints

Neuro-Symbolic Warm-Starting (RL + SMT/ILP)

- Instead of viewing RL and SMT/ILP as competitors, use the sub-second DRL policy (0.038 s) to generate a high-quality initial feasible solution (**warm-start** / upper bound) for an exact OMT or ILP solver, drastically pruning its branch-and-bound tree.

Graph-Based Social & Conflict Constraints

- Incorporate student-to-student pairwise constraints (*"cannot work together"* / *"complementary roles"*) and personality traits (e.g., Belbin or Big Five profiles) into the Dynamic Action Mask.
- Online continual adaptation based on post-project peer feedback.

Conclusion

Summary of Contributions

- Formulated NP-hard constrained student team composition as a sequential **Safe MDP**.
- Combined **Dynamic Action Masking**, **Safety Overrides**, **Cubic Reward Shaping**, and **DQfD Pre-loading** to enforce strict pedagogical bounds.
- Achieved a **204× total speedup** (< 0.04 s inference) and a **20% improvement in inter-group fairness** over iterative OMT (Z3).

Paper & Code Resources:

Proceedings of ICAI 2026: `https://icai.uni-eszterhazy.hu/2026/downloads/icai2026-proceedings.pdf`

GitHub Repository: `https://github.com/herbakmarcell/herbak-icai-2026-drl-student-grouping`

Thank you for your attention!

References

Adil, A. A., Kovásznai, G., & Khalid, M. (2025). Automated fair team formation in STEAM activities using satisfiability modulo theories (SMT). *Annales Mathematicae et Informaticae*, *61*, 1–14. https://doi.org/10.33039/ami.2025.10.001

Alberola, J. M., del Val, E., Sanchez-Anguix, V., Palomares, A., & Teruel, M. D. (2016). An artificial intelligence tool for heterogeneous team formation in the classroom. *Knowledge-Based Systems*, *101*, 1–14. https://doi.org/10.1016/j.knosys.2016.02.010

Baer, J. (2003). Grouping and achievement in cooperative learning. *College Teaching*, *51*(4), 169–175. https://doi.org/10.1080/87567550309596434

Barrett, C., Fontaine, P., & Tinelli, C. (2025). *The SMT-LIB standard: Version 2.7* [Technical report]. Department of Computer Science, The University of Iowa. https://smt-lib.org

Danisovszky, M., Yang, Z. G., & Kusper, G. (2020). Classification of SAT problem instances by machine learning methods. In *Proceedings of the 11th International Conference on Applied Informatics* (pp. 94–100). CEUR-WS.org. https://ceur-ws.org/Vol-2650/paper11.pdf

de Moura, L., & Bjørner, N. (2008). Z3: An efficient SMT solver. In *Tools and algorithms for the construction and analysis of systems* (pp. 337–340). Springer. https://doi.org/10.1007/978-3-540-78800-3_24

Felder, R. M., & Brent, R. (1994). *Cooperative learning in technical courses: Procedures, pitfalls, and payoffs* (ED377038). ERIC. https://eric.ed.gov/?id=ED377038

García, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, *16*, 1437–1480. https://www.jmlr.org/papers/volume16/garcia15a/garcia15a.pdf

Herbák, M. (2026). *herbak-icai-2026-drl-student-grouping* [Computer software]. GitHub. https://github.com/herbakmarcell/herbak-icai-2026-drl-student-grouping

References (continued)

Hester, T., Večerík, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., Dulac-Arnold, G., Agapiou, J., Leibo, J. Z., & Gruslys, A. (2018). Deep Q-learning from demonstrations. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). https://doi.org/10.1609/aaai.v32i1.11757

Hu, X., Chen, P., Wen, Y., Tang, B., & Chen, L. (2024). *Long and short-term constraints driven safe reinforcement learning for autonomous driving* (arXiv:2403.18209). arXiv. https://doi.org/10.48550/arXiv.2403.18209

Masina, G., & Sebastiani, R. (2026). Exploiting partial-assignment enumeration in optimization modulo theories. In *Frontiers of combining systems* (pp. 117–134). Springer. https://doi.org/10.1007/978-3-032-04167-8_7

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Najem, K. (2025). *Student performance and learning behavior dataset for educational analytics* (Version v1.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.16459132

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, *22*(268), 1–8. http://jmlr.org/papers/v22/20-1364.html

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.