Path2Patrol: Path-Conditioned Guard Generation in Stealth Games through MAP-Elites

Yiwei Zhang[1], Kaijie Xu[1], Richard Zhao[2], Clark Verbrugge[1]

[1]School of Computer Science, McGill University, Montréal, Québec, Canada

[2]Department of Computer Science, University of Calgary, Calgary, Alberta, Canada

yiwei.zhang3@mail.mcgill.ca, kaijie.xu2@mail.mcgill.ca, richard.zhao1@ucalgary.ca, clark.verbrugge@mcgill.ca

***Abstract*—Guard patrol behaviors are critical to stealth games. However, existing procedural generation approaches largely focus on the problem of constructing a valid stealth scenario with guard behaviors rather than on creating a scenario that matches a designer's intended player solution. To address this, we propose a novel, designer-centric Path2Patrol (P2P) framework that automates the procedural generation of guard patrols through a Quality Diversity algorithm. Given a designer-specified player solution path, P2P generates reasonable patrols that guarantee the feasibility of the intended trajectory. We integrate three interpretable behavioral dimensions, Coverage, Risk, and Topological Control into MAP-Elites. Our framework not only produces valid, distinct guard patrols but also empowers designers to modulate level difficulty and restrictions via intuitive parameter tuning. Evaluations across four maps against four baselines show our method outperforms existing approaches in quality, diversity, and controllability. As an early effort to characterize player experience through guard authoring from a designer-centric viewpoint, we hope P2P inspires further exploration of experience-driven NPC behavior generation in stealth games.**

***Index Terms*—Stealth games, Procedural content generation, Guard patrol generation, Quality diversity, MAP-Elites**

I. Introduction

Stealth games center on the core interactions between players and patrolling guards; therefore, it is crucial to plan guard patrol routes based on the desired player experience. In current commercial games, designers typically rely on either manually crafting routes supplemented by simple interactive behaviors upon detection, or utilizing AI-driven guard state machines. However, the former manual approach is time-consuming, labor-intensive, and difficult to quantitatively test or evaluate. The latter, while systemic, fails to make the player's traversal and expectations controllable. Under both paradigms, level designers struggle to control players' experience and their intended path through the environment.

To address these limitations, various procedural guard generation methods have been explored. Most prior approaches rely on static geometric map analysis, such as Voronoi diagrams [1] and skeleton roadmaps [2], alongside deterministic algorithms like heuristic search and potential fields [3]. These methods generally generate guard patrols first, where player paths are typically computed merely for post-generation evaluation rather than acting as a foundational prerequisite. Such techniques can support games that focus on random or emergent player solutions, but remain confined to a one-sided layout focused solely on guards. By adhering to this "guard-first" paradigm, it may be hard to deliver desired play experiences in the resulting levels. Smith's GDC talk [4] motivates the problem of generating diverse and controllable guard patrol routes based on a designer's pre-specified player path in stealth games. However, this remains an under-investigated problem.

To bridge these gaps, we propose Path2Patrol (P2P), a novel procedural guard patrol generation framework powered by MAP-Elites [5]. The framework enforces the predefined player path as a hard constraint, ensuring that the generated guard patrols constrict the player's feasible region and viable route alternatives, collapsing the space of safe traversals toward the designer's intended trajectory. Three interpretable behavioral dimensions, namely Coverage, Risk, and Topological Control, are integrated into MAP-Elites and governed by designer-tunable parameters. By adjusting these parameters, designers can shape how aggressively guard patrols compress the player's spatial freedom, achieving diverse and controllable patrol behaviors conditioned on an arbitrary player path.

We evaluate our approach across multiple game environments, player paths, and guard strategies, confirming significant improvements over baselines in both patrol realism and player experience controllability. Our key contributions are:

- A novel, designer-centric Path2Patrol (P2P) framework for procedurally generating diverse, high-quality guard behaviors while maintaining player path feasibility.
- Three interpretable behavioral dimensions (coverage, risk, and topological control) integrated into MAP-Elites to illuminate the search space.
- An intuitive, controllable mechanism for parameter-tuning and elite-selection that enables precise control over level difficulty and restrictions.
- Detailed experimental analyses and novel restriction-centric metrics that validate superior quality and diversity over traditional baselines in complex environments.

II. Related Work

Guard patrol behavior is fundamental to realism, challenge, and immersion in stealth games. Prior procedural generation work is largely guard-first: patrol routes are generated upfront, while player stealth paths are introduced later mainly for evaluation. Xu et al. utilize Voronoi diagrams for level partition and generate rhythm-based patrol routes [1], [6]. Al Enezi et

979-8-3315-9476-3/26/$31.00 ©2026 IEEE

al. employ a staleness-driven model [7] and skeleton graph [2] that drive guards toward unexplored areas. Xu et al. design a composite potential field to produce real-time patrol behavior balancing coverage and pursuit [3]. Despite strong movement planning, the patrols may not align with the intended pacing.

Procedural content generation (PCG) conditioned on player paths or action traces is well established across many game genres. Cooper et al. present a constraint-based method that produces levels for Cave and Mario from designer-specified player routes [8]. Mixed-initiative tools like Tanagra [9], and grammar-based architectures such as Launchpad [10] have been successfully deployed to construct 2D platformer levels from guaranteed player action sequences. Summerville et al. and Sorochan et al. train Long Short-Term Memory (LSTM) networks on player traces to generate levels for Mario and Lode Runner, respectively [11] [12]. These studies suggest that designer-centric PCG with player traces can ensure playability, while also shaping coherent structure and pacing.

While PCG with player paths has been widely studied, its application to patrol generation in stealth games remains under-explored. Smith's GDC talk emphasizes the problem though, outlining a design framework to construct stealth levels from the orchestration of player traversal as well as enemy placement and other factors [4]. Bonnot et al. also explore this direction, proposing an algorithmic approach that derives guard patrols from a player path by sampling and connecting anchor points around it, creating high risk scenarios for the player [13]. Their approach, however, only uses A* for player pathing and does not guarantee viability once guards are placed. We include an adaptation of this approach in our study as a baseline, extending it to support varied paths while preserving their feasibility, although we note that this strategy still struggles to produce diverse patrol behaviors and manage multi-objective optimization.

Since our challenge requires us to satisfy multiple objectives while producing diverse valid patrols rather than a single optimal solution, Quality Diversity algorithms, particularly MAP-Elites [5], offer a suitable paradigm. By maintaining an archive of high-quality, behaviorally distinct elites, MAP-Elites illuminates a multi-dimensional behavioral space and has been successfully applied to PCG across various game content domains, from Mario and dungeon levels [14], [15] to bullet-hell attack patterns and action-adventure enemies [16], [17]. Building on these successes, we propose a path-conditioned, designer-centric framework that leverages MAP-Elites to illuminate the guard behavior search space and procedurally generate guard patrols. The framework yields controllable, diverse, and high-quality patrols spanning combinations of spatial coverage, player-facing risk, and topological control, while guaranteeing that the predefined player path remains a feasible solution to complete the level.

III. Methods

This section defines the grid-based stealth environment, agent dynamics, and path-conditioned guard generation.

A. Environment Abstractions

We model the level as a discrete grid map $\mathcal{M}$ of size $W \times H$, partitioned into free cells $\mathcal{C}_{free}$ and obstacles $\mathcal{C}_{obs}$, where $\mathcal{C}_{free} \cup \mathcal{C}_{obs} = \mathcal{M}$ and $\mathcal{C}_{free} \cap \mathcal{C}_{obs} = \emptyset$. Agents occupy one cell and, at each tick, may wait or move to one of the four orthogonal neighbors, provided the destination lies in $\mathcal{C}_{free}$.

The player trajectory is given as a fixed input sequence over a horizon T: $P = \{p_0, p_1, \ldots, p_T\}$, where $p_t = (x_t^p, y_t^p) \in \mathcal{C}_{free}$. This pre-defined path serves as the foundational reference for all subsequent guard generation and evaluation.

We consider N guards $\mathcal{G} = \{G_1, G_2, \ldots, G_N\}$. Each guard G_i follows a periodic reciprocating patrol to reflect the rhythmic patterns typical of stealth games, enabling players to exploit temporal gaps rather than facing unpredictable return paths. The base one-way route is $\mathcal{R}_i = (r_{i,0}, r_{i,1}, \ldots, r_{i,L-1})$, where the route length satisfies $L \leq L_{max}$. The full spatio-temporal trajectory $\mathbf{g}_i = \{g_{i,0}, g_{i,1}, \ldots, g_{i,T}\}$ is generated by traversing $\mathcal{R}_i$ back and forth.

Each guard has a symmetric field of view with radius r_{fov} (default: 2). At any time step t, a walkable cell is considered unsafe if it lies within Euclidean distance r_{fov} from the guard's current position $g_{i,t}$. The set of visible cells at time t is

$$\mathcal{V}(g_{i,t}) = \{c \in \mathcal{C}_{free} \mid d(g_{i,t}, c) \leq r_{fov}\}.$$

To ensure the provided player path P remains strictly feasible, guards must never observe the player at any time step. This imposes the hard constraint:

$$\forall t \in \{0, \ldots, T\}, \forall i \in \{1, \ldots, N\}: \quad p_t \notin \mathcal{V}(g_{i,t})$$

Our goal is to generate diverse patrol routes $\{\mathcal{R}_1, \ldots, \mathcal{R}_N\}$ (and their induced trajectories $\{\mathbf{g}_1, \ldots, \mathbf{g}_N\}$) that satisfy this hard visibility constraint, yielding a portfolio of valid stealth guards consistent with the intended player flow.

B. MAP-Elites

MAP-Elites drives our path-conditioned guard generation and multi-objective optimization by illuminating a multi-dimensional behavioral space with high-quality elites. It yields a diverse, valid archive of patrol routes, providing designers with a rich portfolio of difficulty and strategic profiles.

1) ***General Architecture****:* To avoid the combinatorial complexity of jointly evolving N guards, we adopt a sequential scheme with N MAP-Elites passes. Let $\mathcal{G} = \{G_1, \ldots, G_N\}$ be the guard set. The i-th pass evolves G_i conditioned on the static map $\mathcal{M}$, the fixed player path P, and the already-committed routes of previous guards $\mathcal{H}_{i-1} = \{\mathbf{g}_1, \ldots, \mathbf{g}_{i-1}\}$.

2) ***Genotype to Trajectory****:* We use a compact genotype to reduce search dimensionality. With the start cell $s_i \in \mathcal{C}_{free}$ fixed by a random placement policy, an individual is a variable-length waypoint list $\mathbf{w} = [w_1, \ldots, w_m]$, where $w_j \in \mathcal{C}_{free}$ and $m \in [m_{min}, m_{max}]$. The phenotype is the base patrol route $\mathcal{R}_i$, produced by connecting $(s_i \to w_1 \to \cdots \to w_m)$ using a time-dependent constrained A* that rejects expansions causing conflicts with P or $\mathcal{H}_{i-1}$ at the corresponding time step. If the accumulated route exceeds L_{max}, it is truncated. The full reciprocating trajectory $\mathbf{g}_i$ is then obtained by repeating $\mathcal{R}_i$ over the horizon T via the periodic mapping defined in Section III-A, ready for subsequent evaluation.

*3) **Behavioral Dimensions**:* MAP-Elites indexes individuals by behavioral dimensions (BDs) independent of fitness. We define three BDs to characterize guard patrols: incremental spatial coverage, player-facing risk, and topological control.

*a) **Coverage** (BD_{cov}):* Overall level coverage can be expressed as Global Map Coverage Metric (M_{cov}), the fraction of walkable cells observed by guards over time:

$$M_{cov} = \frac{1}{|\mathcal{C}_{free}|} \left| \bigcup_{i=1}^{N} \bigcup_{t=0}^{T} \mathcal{V}(g_{i,t}) \right|.$$

However, during the i-th sequential pass, a single guard typically contributes a small and highly clustered absolute coverage. We therefore measure *incremental* coverage: the number of newly observed cells contributed by G_i beyond what previous guards already cover, normalized by a capacity constant C_{max} (the maximum number of distinct cells a guard could observe under the patrol-length limit). Let $\Omega_{seen}^{(i-1)}$ denote the union of cells already monitored by guards $1, \ldots, i-1$:

$$BD_{cov}^{(i)} = \frac{1}{C_{max}} \left| \left(\bigcup_{t=0}^{T} \mathcal{V}(g_{i,t}) \right) \setminus \Omega_{seen}^{(i-1)} \right|.$$

This yields a stable descriptor spanning from heavily redundant patrols to largely disjoint additions. BD_{cov} quantifies the novel spatial area covered by the guard's trajectory.

*b) **Risk** (BD_{risk}):* Adapted from Tremblay et al., 2014 [18], a robust risk metric is cumulative inverse distance to the player along the path, using shortest-path distance d^*:

$$M_{risk} = \sum_{t=0}^{T} \sum_{i=1}^{N} \frac{1}{d^*(p_t, g_{i,t})^3}.$$

In actual gameplay, high risk moments are sparse, as a guard typically approaches the player only during brief action windows. Despite M_{risk} being a popular metric for measuring risk, standard accumulation or averaging dilutes brief but critical high risk windows across a long trajectory, making it unsuitable as the behavioral descriptor.

Instead, our descriptor operates in two stages. We first define the instantaneous risk u_t for the guard g_i at every step t using a Gaussian formula ($\sigma_{risk} = 2$) to ensure a smooth decay.

$$u_t = \exp\left(-\frac{(d^*(g_{i,t}, p_t) - r_{fov})^2}{2\sigma_{risk}^2} \right).$$

Then we aggregate these sequential instantaneous risks with a Log-Mean-Exp operator. This soft-maximum approach ($\beta = 10$) elegantly bounds the dimension to $[0, 1]$ while emphasizing key moments of high risk across the trajectory.

$$BD_{risk}^{(i)} = \frac{1}{\beta} \log \left(\frac{1}{T+1} \sum_{t=0}^{T} e^{\beta \cdot u_t} \right).$$

BD_{risk} is bounded and well distributed, allowing the MAP-Elites to illuminate the risk dimension rather than collapsing into few cells. A higher value indicates a higher comprehensive risk of the guard to the player throughout the game.

*c) **Topological Control** (BD_{topo}):* Controlling topological chokepoints (e.g., intersections and narrow corridors) has high-level strategic values. To quantify this, we compute the Betweenness Centrality (BC) via Brandes' algorithm on the walkable graph. Topological Control Metric (M_{topo}) is computed as the mean BC per guard's monitored area (Ω_i), averaged across all N guards. High M_{topo} indicates the guards are patrolling topologically critical regions.

$$M_{topo} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{|\Omega_i|} \sum_{c \in \Omega_i} BC(c) \right)$$

Since raw BC produces sharp spikes that create a rugged optimization landscape, we apply a 2D Gaussian smoothing kernel (kernel size 5×5, $\sigma_{smooth} = 1.0$) to the raw BC values, resulting in a smoothed spatial map $\tilde{B}(c)$.

The individual topology descriptor, $BD_{topo}^{(i)}$, is then defined as the average smoothed BC within guard i's field of view ($\mathcal{V}(g_{i,t})$) over the episode (T):

$$BD_{topo}^{(i)} = \frac{1}{T+1} \sum_{t=0}^{T} \left(\frac{1}{|\mathcal{V}(g_{i,t})|} \sum_{c \in \mathcal{V}(g_{i,t})} \tilde{B}(c) \right)$$

Figure 1 shows the raw and smoothed BC in a game level. This transformation allows the algorithm to guide guards smoothly. BD_{topo} assesses the structural importance of the guard's patrol zone within the global map topology. A high value indicates that a guard monitors the map's vital topological transit points.

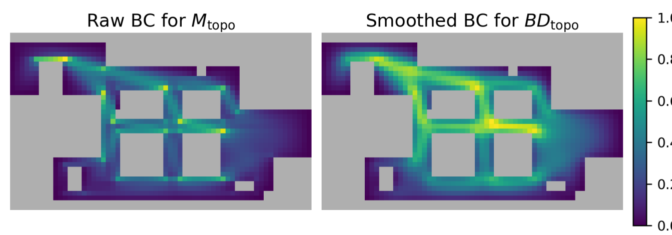

Fig. 1: Normalized Raw BC and Smoothed BC Heatmaps

*4) **Evolutionary Process**:* MAP-Elites, a variant of the standard genetic algorithm, inherits the conventional evolutionary phases, including initialization, evaluation, crossover, and mutation operators.

*a) **Initialization and Evaluation**:* The initial archive is populated by N_{init} randomly generated individuals, whose genotypes comprise m sampled valid waypoints ($m \in [m_{min}, m_{max}], m_{min} = 3, m_{max} = 10$). Upon mapping these genotypes to complete guard trajectories (Section III-B2), candidates must be filtered by a full-trajectory feasibility check. Any individual intersecting the player is immediately discarded. Feasible guard trajectories are assigned three behavioral dimensions (BD_{cov}, BD_{risk}, BD_{topo}) and evaluated by a kinematic smoothness fitness f. Serving as a tie-breaker, f evaluates the kinematic smoothness and penalizes directional shifts (N_{turns}) along a route $\mathcal{R}_i$ of length L:

$$f(\mathcal{R}_i) = 1 - \frac{N_{turns}}{\max(L-2, 1)}$$

*b) **Archive Update and Iteration**:* The BD space ($[0, 1]^3$) is uniformly discretized into 1,000 bins (step size 0.1). When a feasible individual is evaluated, its BDs determine its corresponding bin index. It enters the archive if its mapped bin is vacant, or if its fitness f strictly exceeds the current elite.

Parents are selected from the archive and N_{pop} (=100) offspring are generated per iteration using the following variation:

- **Mutation (50%):** Structural genotype changes applied via equiprobable waypoint deletion, insertion, spatial perturbation, or complete resampling.
- **Crossover (50%):** A single-point crossover concatenates the front segment of parent A's waypoints with the back segment of parent B's. It degrades into mutation if the connection distance is greater than 5 (default threshold).

*c) **Final Elite Selection**:* Once the evolutionary loop for guard G_i terminates, the algorithm selects a desired guard trajectory g_i^* from the archive as the final result. To ensure guard quality, candidates are pre-filtered according to the 3-sigma rule ($f \geq \mu_f - 3\sigma_f$). The final deployed individual maximizes a weighted sum of normalized behavioral descriptors:

$$U_{elite}(g_i) = w_1 \widehat{BD}_{cov} + w_2 \widehat{BD}_{risk} + w_3 \widehat{BD}_{topo}$$

where $\widehat{BD}$ denotes the min-max normalized values within the archive, and the weight vector $w = (w_1, w_2, w_3)$, each element $w_i \in [-1, 1]$, is a designer-defined weight reflecting the desired characteristics of the level. A positive weight indicates the degree of preference for maximizing a specific dimension, whereas a negative weight encourages minimizing it. Ties are broken by the intrinsic smoothness fitness f.

IV. Experiments & Results

This section outlines the experiments for evaluating our MAP-Elites framework, including the settings, baselines, and a quantitative result analysis of the stealth guards.

A. Scenario Details

*1) **Game Levels and Player Paths**:* We evaluate our methods on four grid-based levels from *Metal Gear Solid* (MGS), *Dragon Age: Origins* (den207d, den101d), and *Hitman 2 Miami*, summarized in Table I. MGS and Hitman are used in prior research [1], [3], while the Dragon Age levels are selected from the *movingai* dataset [19] as similar-sized levels.

Our framework allows designers to input any arbitrary player path. For fair evaluations, we automatically generate two classes of unique player trajectories: *shortest* and *long* paths (25 each, 50 total). Shortest paths include an A* path of length D^* along with alternative shortest paths enumerated by bidirectional BFS distance maps. Long exploratory trajectories are produced with an A*-variant that weakens the heuristic and penalizes revisits to encourage detours; lengths are constrained to $[1.1, 2.0] \times D^*$ and augmented with random wait steps to reflect human play. Figure 2a shows all player paths in MGS.

Guards are then initialized randomly on feasible cells. Each guard starts at least r_{fov} from the player and at least $2 \times r_{fov}$ from other guards to avoid immediate overlap. All methods are evaluated from the same initial configurations to ensure fair comparison, except the algorithmic baseline, which follows its own guard placement scheme.

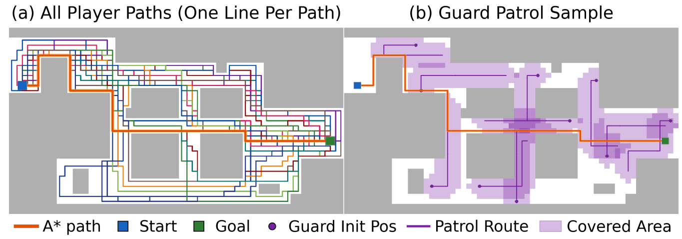

Fig. 2: All Player Paths and Guard Patrol Sample

Level Name	Grid Properties		
	Dimensions (Rows × Cols)	Walkable (Count (%))	Obstacles (Count (%))
MGS	37 × 63	1006 (43%)	1325 (57%)
Origins den207d	50 × 38	874 (46%)	1026 (54%)
Origins den101d	41 × 73	1360 (45%)	1633 (55%)
Hitman Miami	52 × 42	1492 (68%)	692 (32%)

TABLE I: Characteristics of the four levels

*2) **Methods**:* We have four interpretable baselines.

- **Algorithmic**: Adapted from Bonnot's work [13], this is a deterministic baseline that samples two patrol anchors on opposite sides of the player path with random perturbations (within 15 cells), and connects them with constrained A* (Section III-B2) to form the reciprocating patrol route, subject to length limit L_{max}.
- **Random Walk**: At each tick, the guard moves to a random valid neighbor which satisfies the hard constraint.
- **Naive GA**: The same evolutionary pipeline as our method, but run as a standard genetic algorithm optimizing only the fitness f without behavioral dimensions.
- **Greedy**: A greedy search planner with backtracking. Starting from a valid point with at least one valid neighbor, it iteratively adds feasible steps and chooses candidates with the highest composite utility:

  $$U(x_k) = U_{cov}(x_k) + U_{risk}(x_k) + U_{topo}(x_k),$$

  where U_{cov} calculates the average inverse visit count within the candidate's FOV to penalize redundant patrolling, $U_{risk} = \exp(-(d^*(g_{i,t}, p_t) - r_{fov})^2 / 2\sigma_{risk}^2)$ biases towards the player, and U_{topo} rewards patrols with higher smoothed betweenness centrality $\tilde{B}(c)$.
- **±MAP-Elites (Ours)**: To show the upper and lower performance bounds, we instantiate MAP-Elites under two extreme parameter configurations: +MAP-Elites ($w = [+1, +1, +1]$) selects elites maximizing all normalized descriptors, whereas −MAP-Elites ($w = [-1, -1, -1]$) selects elites minimizing them.

All guards generated by each method successfully satisfy the hard constraints: they do not detect the player, and their one-way patrol length does not exceed L_{max}. Figure 2b shows a guard patrol sample by +MAP-Elites given an A* path.

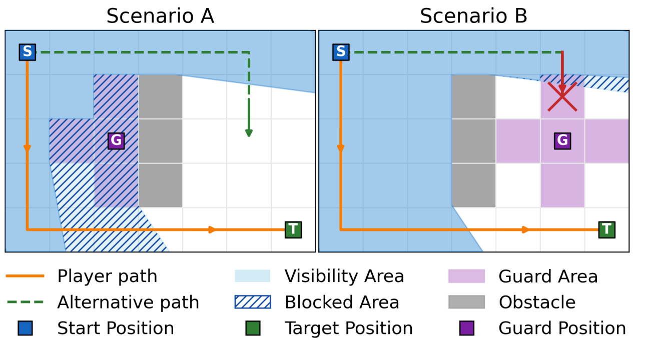

Fig. 3: Two Stealth Scenarios Explaining Restriction Metrics

Scenario	M_{perc}	V_{static}	V_{guard}	M_{obj}
A	**0.3098**	30	17	0.2467
B	0.1857	30	15	**0.3010**

TABLE II: Summary of Restriction Metrics in Scenario A, B

B. *Metrics*

We evaluate the generated guards using six metrics that capture spatial coverage, player-facing risk, topological control, perceived and objective restrictions, and patrol smoothness.

*1) **Coverage, Risk And Topological Control**:* While the Behavioral Dimensions guide per-guard evolution, guard performance is evaluated using corresponding behavioral objective metrics: Coverage (M_{cov}), Risk (M_{risk}), and Topological Control (M_{topo}). Each describes guard influence over time and all guards, providing a direct measure of how much space is monitored, how closely the squad pressures the player path, and how strongly it controls topologically important regions. Higher values indicate a more challenging stealth level.

*2) **Perceived Restriction through Isovist** (M_{perc}):* Placing guards that respect and enforce a given player path can overly restrict a player's sense of agency. We quantify the player's perceived restriction using continuous Isovists [20]. M_{perc} quantifies the average relative reduction in visually safe area caused by guards, compared to a static-obstacle baseline:

$$M_{perc} = 1 - \frac{1}{T+1} \sum_{t=0}^{T} \frac{\text{Area}(\mathcal{I}_{guard}(p_t))}{\text{Area}(\mathcal{I}_{static}(p_t))}$$

where $\mathcal{I}_{static}$ is the player's visible area without guards and $\mathcal{I}_{guard}$ represents the remaining area when guards are present. Higher values reflect greater psychological restriction.

*3) **Objective Restriction via Path Counting** (M_{obj}):* Since isovist analysis overlooks restriction imposed by unseen guards, we quantify the objective restriction by computing the goal-directed path volume via Dynamic Programming on a time-expanded graph (4-neighbor plus waiting) [21]. Let $V_t(c)$ represent the number of safe paths reaching cell c at time t:

$$V_t(c) = \mathbf{1}_{safe}(c,t) \sum_{c' \to c} V_{t-1}(c')$$

where $\mathbf{1}_{safe}(c,t)$ ensures c is free from obstacles and guard vision. Summing paths reaching the goal at T gives the total safe path counts with (V_{guard}) and without (V_{static}) guards. M_{obj} is defined logarithmically to reflect exponential path compression. A higher value denotes stricter restriction:

$$M_{obj} = -\log_{10}\left(\frac{V_{guard}}{V_{static}}\right)$$

For instance, Fig. 3 and Table II contrast perceived restriction with objective restriction. In Scenario A, a visible guard ($r_{fov} = 1$) substantially shrinks the player's isovist, increasing perceived pressure while preserving more alternative routes (high M_{perc}, low M_{obj}). In Scenario B, an out-of-sight guard exerts little visual threat but blocks critical corridors, sharply reducing viable paths (low M_{perc}, high M_{obj}). This shows that evaluating restriction via path counting captures structural restrictions extending far beyond the player's field of view.

*4) **TurnCount** (M_{turns}):* It quantifies the number of directional changes that the guard makes (N_{turns}) during a single one-way patrol pass, assessing the behavioral smoothness. Lower values indicate straighter, more realistic patrol patterns.

C. *Guard Properties and Evolutionary Analysis*

We first examine how the guard squad size N and maximum patrol length $L_{\max}$ influence the difficulty of generated stealth levels. Using the MGS map and all player trajectories, we run +MAP-Elites with $w = [1,1,1]$ for 1000 MAP-Elites iterations while sweeping $N \in [2,16]$ with step size 2, and $L_{\max} \in [5,40]$ with step size 5. Metrics are reported as averages over 500 independent trials per configuration.

Figure 4 shows that more guards and longer patrol lengths increase level difficulty. When scaling up N and $L_{\max}$, we observe higher coverage, risk and restrictions. Notably, topological control exhibits a different pattern: per-guard centrality M_{topo} often peaks for smaller squads with long routes, because patrols concentrate around important areas rather than spreading widely. Given the similar dimensions of the four maps, we adopt a default configuration of $N = 10$ and $L_{\max} = 20$ for all subsequent experiments to balance quality and cost.

We also analyze the evolutionary process of MAP-Elites. Figure 5 reports number of elites and fitness statistics, averaged across all 10 guards. Fitness improves rapidly and stabilizes early. The archive population grows sharply during the initial phase, but it slows substantially after 1000 iterations. We therefore adopt 1000 MAP-Elites iterations as the default for all experiments, requiring about 80 seconds per guard.

D. *Performance Analysis: MAP-Elites and Baselines*

To assess the overall effectiveness of our method, we perform a large-scale, aggregated performance study that contrasts the two extreme configurations of our MAP-Elites framework with four baselines. Experiments are conducted on all four game levels, and 50 player paths per level. Each algorithm repeats 500 runs per player path. All metrics are aggregated and averaged over these runs and levels.

Aggregated results in Table III show the inherent limitations of the baseline methods: Naive GA has strong coverage and smooth patrol but neglects risk and topological control, yielding low-tension patrols. The Algorithmic method prioritizes

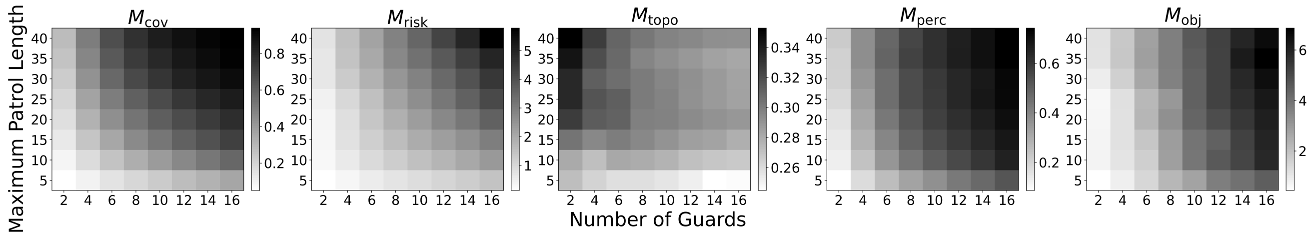

Fig. 4: Five behavioral and restriction metric heatmaps for different number of guards and maximum patrol length.

		Behavioral Metrics			**Restriction Metrics**		**Patrol Smoothness**
GuardMode	PlayerPath	$M_{cov}(\%)\uparrow$	$M_{risk}\uparrow$	$M_{topo}\uparrow$	$M_{perc}(\%)\uparrow$	$M_{obj}\uparrow$	$M_{turns}\downarrow$
− MAP-Elites	Short	26.72 ± 5.21	$\mathbf{0.19 \pm 0.04}$	$\mathbf{0.09 \pm 0.04}$	$\mathbf{20.26 \pm 5.31}$	$\mathbf{0.35 \pm 0.33}$	2.03 ± 0.07
RandomWalk	Short	36.87 ± 6.83	0.83 ± 0.20	0.14 ± 0.06	39.95 ± 8.24	1.97 ± 0.80	$\mathbf{14.31 \pm 0.11}$
Algorithmic	Short	$\mathbf{16.65 \pm 2.33}$	2.01 ± 0.18	$\mathbf{0.24 \pm 0.08}$	26.66 ± 5.53	1.12 ± 0.90	2.62 ± 1.37
Naive GA	Short	52.66 ± 7.98	0.64 ± 0.22	0.15 ± 0.06	40.21 ± 11.52	1.48 ± 0.87	$\mathbf{0.86 \pm 0.14}$
Greedy	Short	47.48 ± 7.61	1.58 ± 0.13	0.20 ± 0.07	56.27 ± 7.50	3.59 ± 0.41	9.66 ± 0.26
+ MAP-Elites	Short	$\mathbf{53.68 \pm 8.57}$	$\mathbf{2.05 \pm 0.21}$	0.22 ± 0.05	$\mathbf{59.95 \pm 7.65}$	$\mathbf{3.90 \pm 1.02}$	2.31 ± 0.05
− MAP-Elites	Long	$\mathbf{27.36 \pm 5.79}$	$\mathbf{0.27 \pm 0.06}$	$\mathbf{0.10 \pm 0.04}$	$\mathbf{21.52 \pm 4.68}$	$\mathbf{0.88 \pm 0.53}$	2.19 ± 0.19
RandomWalk	Long	37.14 ± 6.75	1.02 ± 0.23	0.14 ± 0.06	38.50 ± 6.82	2.83 ± 0.70	$\mathbf{14.22 \pm 0.15}$
Algorithmic	Long	29.47 ± 7.65	1.69 ± 0.34	$\mathbf{0.23 \pm 0.07}$	38.89 ± 4.97	3.38 ± 2.57	2.59 ± 0.75
Naive GA	Long	52.78 ± 7.60	0.67 ± 0.22	0.16 ± 0.07	37.32 ± 9.20	2.88 ± 1.36	$\mathbf{0.89 \pm 0.14}$
Greedy	Long	47.52 ± 7.32	1.50 ± 0.39	0.20 ± 0.07	48.99 ± 9.27	5.27 ± 1.47	9.70 ± 0.30
+ MAP-Elites	Long	$\mathbf{55.14 \pm 9.46}$	$\mathbf{1.94 \pm 0.35}$	0.21 ± 0.06	$\mathbf{54.45 \pm 7.07}$	$\mathbf{5.53 \pm 1.28}$	2.06 ± 0.13

TABLE III: Summary of all metrics from each method. All values are mean ± std (two decimals). We highlight the maximum and minimum values for all metrics within each PlayerPath category to show the hardest and easiest cases. ↑: higher values indicate more difficult guard patrols. ↓: lower values indicate straighter patrols.

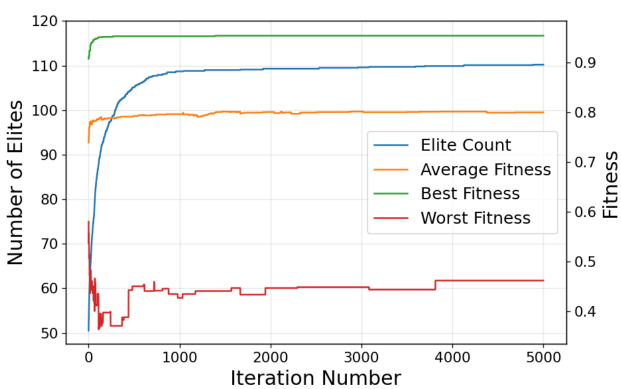

Fig. 5: MAP-Elites Evolutionary Process over 5000 Iterations

high-risk guards at the expense of coverage and restriction. While Greedy search emerges as the most competitive baseline, its preference for immediate gains makes it likely to get trapped in local optima, resulting in patrols with more turns and inferior overall performance compared to MAP-Elites.

In contrast, our designer-centric MAP-Elites framework effectively spans and balances these behavioral dimensions. The difficulty-maximizing +MAP-Elites consistently achieves the most difficult levels with the highest global coverage, risk, and restrictions. Conversely, the −MAP-Elites produces the most conservative guard behaviors. Ultimately, these two configurations show that our approach achieves superior and more diverse performance than baselines, providing designers with a broad range of level designs and difficulty profiles.

E. Controllability Experiment

To assess controllability beyond ±MAP-Elites, we sweep designer weights (w_1, w_2, w_3) on each behavioral dimension. Under the standard setting (N=10, 1000 iterations, all paths, 500 repeats), we conduct three univariate parameter sweep analyses on the MGS level: in each sweep, one weight varies from -1 to 1 in increments of 0.1, while the other two are fixed at either 1 or -1.

Figure 6 reports main effects (the metric aligned with the swept weight), cross effects (remaining behavioral metrics), and restriction effects (M_{perc}, M_{obj}). Across all sweeps, the main effect is strictly monotonic: increasing a dimension's weight increases its corresponding metric, directly validating controllability. Cross effects exhibit mild, non-linear variation, indicating that the dimensions are largely decoupled, albeit with a weak interdependence; given that dimensions are not perfectly orthogonal, these fluctuations are acceptable. For restriction, larger weights generally induce higher restriction. Adjusting BD_{risk} and BD_{topo} weight produces a larger impact on restriction than adjusting coverage. This suggests a design implication: coverage alone is not enough; patrols that emphasize risk and topological influence better shape guard-player interactions.

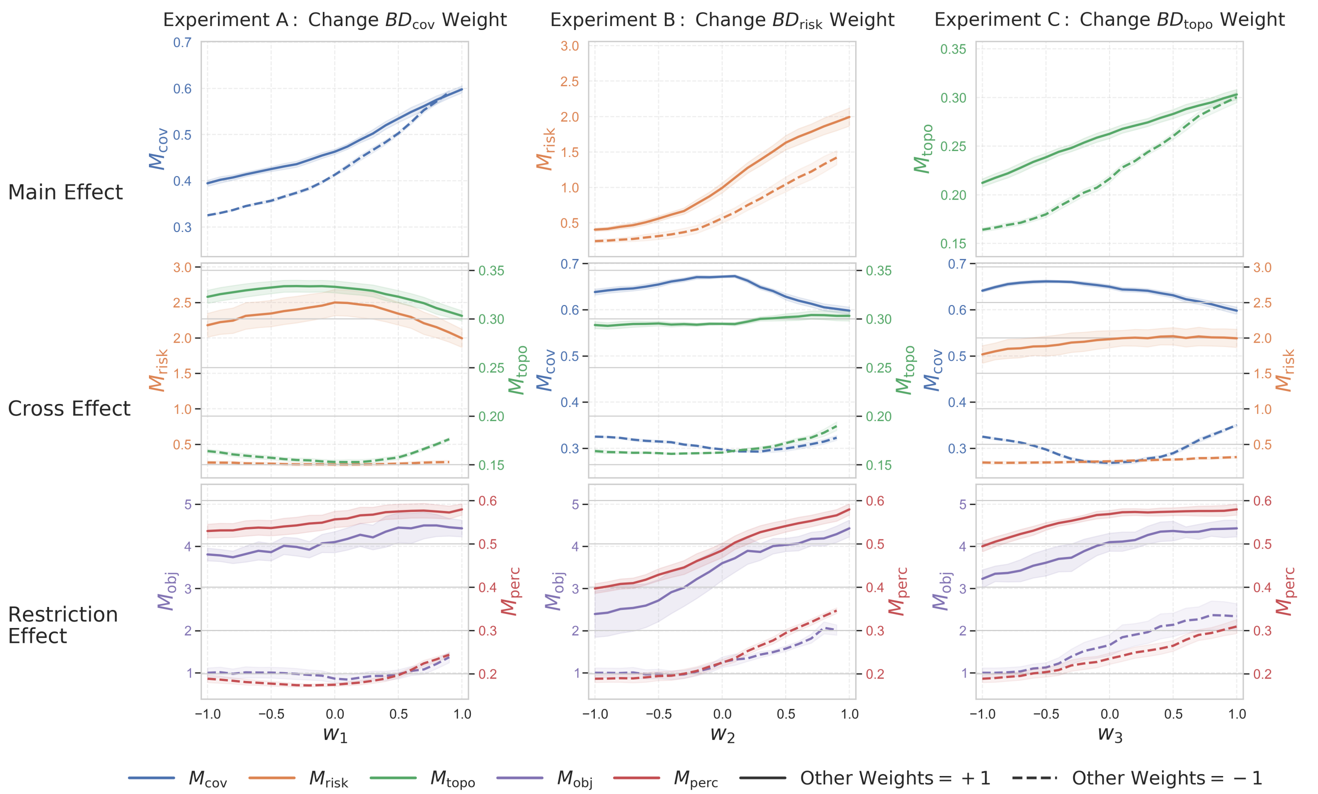

Fig. 6: Five evaluation metrics across weight sweeps. Solid lines: other weights fixed at 1; Dashed lines: others fixed at -1.

Overall, our experiments show that this framework has precise and strong control over guard behaviors, facilitating a wide range of patrol styles from lenient to highly challenging.

F. MGS Case Study: Guard Route Heatmap Analysis

We complement the quantitative results with qualitative heatmaps on the MGS level, fixing the player path to the optimal A* path (shown in Figure 2).

Firstly, we conduct a method-comparison experiment across all methods in Section IV-A2. For each method (or weight setting), we aggregate guard trajectories over 500 repetitions into an averaged visit count heatmap. From all these heatmaps in Figure 7, we find that Naive GA and Random Walk yield near-uniform patterns, showing no response to the player trajectory or map structure. The Algorithmic method concentrates around the player, increasing risk while sacrificing global coverage and restrictions. Greedy has similar patterns with +MAP-Elites since it optimizes multi-objective utility, but its myopic decisions cause it to produce jagged routes and fail to concentrate on crossings, indicating it settles into local rather than global optima. In contrast, our +MAP-Elites achieves the most balanced routes: guards have wide coverage while applying pressure near the player and occupying structurally important areas such as the central crossing. As expected, −MAP-Elites is the most conservative, with patrols along the periphery. Overall, these heatmaps illustrate that MAP-Elites can balance multiple behavioral dimensions simultaneously, yielding patrols at both extremes of difficulty.

To observe how different dimensions and weights shape different patrol styles, we conduct a bivariate behavior analysis via a pairwise weight grid study. We fix one weight to zero and iterate the other two over the set $\{-1, 1\}$ in pairwise combinations. Heatmaps in Figure 8 validate our design: emphasizing coverage spreads visits broadly across the map, risk concentrates patrol pressure near the player path, and topology drives patrols toward structurally important regions; negative weights induce the opposite tendencies. All dimensions are largely decoupled, showing minimal interference with one another; thus, designers can compose preferences across three dimensions to obtain mixed styles. Overall, this supports the controllability and practical effectiveness of our framework.

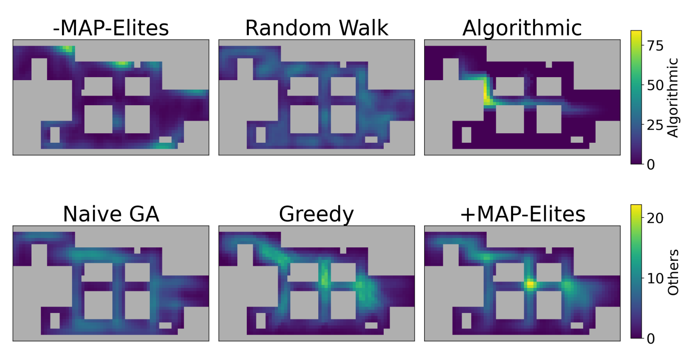

Fig. 7: Method Comparison Visit Heatmaps

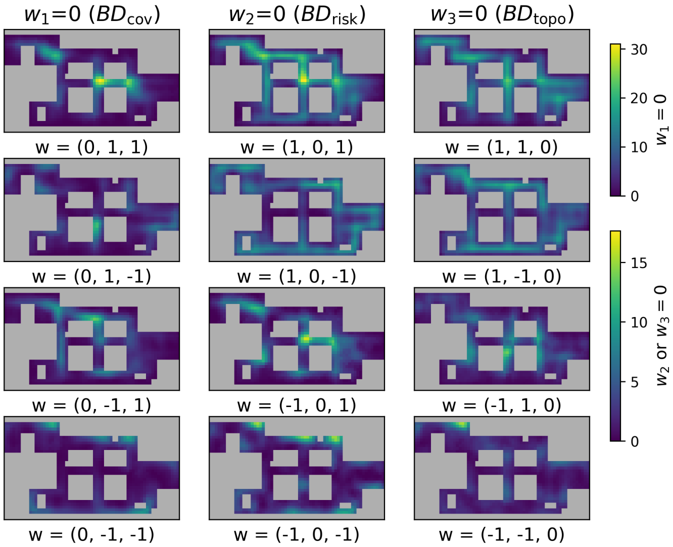

Fig. 8: Bivariate Analysis Visit Heatmaps

V. Conclusion

We present Path2Patrol (P2P), a controllable Quality Diversity framework for procedurally generating guard patrol behaviors conditioned on predefined player trajectories. By embedding three interpretable behavioral dimensions into MAP-Elites, the method guarantees feasible patrols while exposing designer-facing controls for selecting distinct guard playstyles. Across four maps and diverse player paths, our approach consistently produces higher-quality and more diverse patrol sets than four baselines. Targeted controllability studies and visit heatmap analysis further indicate that designers can reliably steer behavioral outcomes through parameter tuning, yielding different guard configurations across multiple dimensions.

Despite the above contributions, our work has limitations. Without a user study, we have not evaluated how our method affects human players and designers. Additionally, our current setting uses a sequential architecture, and assumes simplified guards with reciprocating patrols and symmetric field of view; extending to broader stealth contexts requires additional modeling. Moreover, the guard routes are computed offline and fixed; the current framework does not model the reactive changes in guard behavior once the player is detected, a situation that may arise in real gameplay.

Future work is focused on developing realistic play-testing agents to evaluate generated routes, followed by formal user studies assessing player experience and designer utility. We also plan to scale our approach to more expansive stealth environments with richer guard mechanics, such as multi-state alert behaviors and complex patrol rhythms. Finally, we aim to support real-time generation, enabling the integration of route planning with reactive AI behaviors.

We envision our P2P framework as an important step towards procedural stealth guard generation that adapts to both player agency and explicit design goals.

Acknowledgments: This research supported in part by Natural Sciences and Engineering Research Council of Canada grant RGPIN-2026-07194.

References

[1] Q. Xu, J. Tremblay, and C. Verbrugge, "Generative methods for guard and camera placement in stealth games," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2014, pp. 87–93.

[2] W. Al Enezi and C. Verbrugge, "Skeleton-based multi-agent opponent search," in *IEEE Conference on Games (CoG)*, 2021, pp. 1–8.

[3] K. Xu and C. Verbrugge, "Generic guard AI in stealth game with composite potential fields," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2025, pp. 153–162.

[4] R. Smith, "Level-building for stealth game play," Game Developers Conference (GDC), 2006. [Online]. Available: http://www.roningamedeveloper.com/Materials/RandySmith_GDC_2006.ppt

[5] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites. corr abs/1504.04909 (2015)," *arXiv:1504.04909*, 2015.

[6] Q. Xu, J. Tremblay, and C. Verbrugge, "Procedural guard placement for stealth games," in *Fifth Workshop on Procedural Content Generation in Games*, 2014.

[7] W. Al Enezi and C. Verbrugge, "Dynamic guard patrol in stealth games," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2020, pp. 160–166.

[8] S. Cooper and M. Guzdial, "path2level: Constraint-based level generation from paths," in *IEEE Conference on Games (CoG)*, 2023, pp. 1–4.

[9] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive planning and constraint solving for mixed-initiative level design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 201–215, 2011.

[10] G. Smith, J. Whitehead, M. Mateas, M. Treanor, J. March, and M. Cha, "Launchpad: A rhythm-based level generator for 2-D platformers," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 1, pp. 1–16, 2010.

[11] A. Summerville, M. Guzdial, M. Mateas, and M. Riedl, "Learning player tailored content from observation: Platformer level generation from video traces using LSTMs," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2016, pp. 107–113.

[12] K. Sorochan, J. Chen, Y. Yu, and M. Guzdial, "Generating Lode Runner levels by learning player paths with LSTMs," in *16th International Conference on the Foundations of Digital Games*, 2021, pp. 1–7.

[13] A. Bonnot, Y. Francillette, B. A. Menelas, and B. Bouchard, "Enhancing stealth gameplay through procedural generation: An algorithmic approach to dynamic guard paths and placement in infiltration games," in *IEEE Gaming, Entertainment, and Media Conference*, 2024, pp. 1–6.

[14] M. C. Fontaine, R. Liu, A. Khalifa, J. Modi, J. Togelius, A. K. Hoover, and S. Nikolaidis, "Illuminating Mario scenes in the latent space of a generative adversarial network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 5922–5930.

[15] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, "Empowering quality diversity in dungeon design with interactive constrained map-elites," in *IEEE Conference on Games (CoG)*, 2019, pp. 1–8.

[16] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, "Talakat: Bullet hell generation through constrained map-elites," in *Proceedings of The Genetic and Evolutionary Computation Conference*, 2018, pp. 1047–1054.

[17] B. M. Viana, L. T. Pereira, and C. F. Toledo, "Illuminating the space of enemies through map-elites," in *IEEE Conference on Games (CoG)*, 2022, pp. 17–24.

[18] J. Tremblay, P. A. Torres, and C. Verbrugge, "Measuring risk in stealth games." in *9th International Conference on the Foundations of Digital Games*, 2014, pp. 1–8.

[19] N. Sturtevant, "Benchmarks for grid-based pathfinding," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 2, pp. 144 – 148, 2012.

[20] M. L. Benedikt, "To take hold of space: isovists and isovist fields," *Environment and Planning B: Planning and design*, vol. 6, no. 1, pp. 47–65, 1979.

[21] R. Goldstein, K. Walmsley, J. Bibliowicz, A. Tessier, S. Breslav, and A. Khan, "Path counting for grid-based navigation," *Journal of Artificial Intelligence Research*, vol. 74, pp. 917–955, 2022.