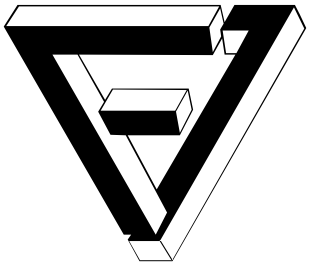
Masaryk University
Faculty of Informatics



Bridging Continuous Learning and Discrete Optimization in Artificial Intelligence

Habilitation Thesis
(Collection of Articles)

Vít Musil

Brno, 2026

Abstract

This thesis presents the author's contribution to the optimization principles that connect continuous learning with discrete decision-making. The central problem is structural: modern systems are trained in differentiable spaces, while many decisions are combinatorial, constrained, and operationally discrete. The thesis addresses this mismatch through mathematically explicit interfaces that preserve both end-to-end trainability during learning and validity at execution.

The first contribution stream concerns differentiable programming for combinatorial algorithms. It introduces principled surrogate-gradient techniques for non-differentiable operators, including black-box optimization modules and discontinuous computational components. These methods enable end-to-end objective-driven training without replacing exact algorithmic solvers with simplified surrogates. The resulting framework improves decision quality in structured tasks where predictive accuracy alone is insufficient.

The second contribution stream concerns strategy synthesis in stochastic and adversarial environments. It formulates finite-memory control synthesis as continuous optimization over randomized strategy spaces, with structured projection back to executable controllers. The thesis develops tractable evaluation procedures for long-run objectives, including recurrent reachability and mean-payoff criteria under local constraints.

Across both streams, the unifying outcome is a coherent methodology for neuro-symbolic and decision-aware AI: continuous optimization is used not as a substitute for discrete reasoning, but as a scalable mechanism to synthesize high-quality discrete decisions under formal constraints.

Acknowledgement

I am deeply grateful to my coauthors Brandon Amos, Tomáš Brázdil, Andreas René Geist, Sebastian Hoffmann, David Klaška, Vojtěch Kůr, Antonín Kučera, Volodymyr Kuleshov, Martin Kurečka, Georg Martius, Claudio Michaelis, Petr Novotný, Anselm Paulus, Marin Vlastelica Pogančić, Simon Rappenecker, Vojtěch Řehák, Michal Rolínek, Subham Sekhar Sahoo, Pierre Schumacher, Paul Swoboda, and Dominik Zietlow. Every result collected here was shaped by collaboration, discussion, and shared intellectual effort. I sincerely thank each of you for your trust, creativity, and commitment.

My sincere thanks belong to Luboš Pick and all the math folks around him. Thanks to them, I gained a deep understanding of mathematics, especially analysis, and, equally importantly, a lasting sense of rigor and a love of long walks.

Special thanks go to Michal Rolínek and to the group of Georg Martius at MPI in Tübingen, who initiated my journey in computer science and strongly influenced its direction. It was always a pleasure to stay at MPI and cover the walls with formulas. I won't forget our jam, table soccer, bouldering, and Boulanger sessions.

I also wish to thank Tony Kučera and all my colleagues at the Faculty of Informatics, Masaryk University, for creating a friendly and flourishing environment. The atmosphere of openness, curiosity, and mutual support has been essential to my work. I would like to thank Tomáš Brázdil for his sense of humor and countless insightful remarks, many of which originated from Yes, Minister. His ability to channel the wisdom of Sir Humphrey has been both enlightening and, at times, deeply concerning.

My most important thanks belong to my wife, Jana, and our sons, Teo and Eda. Your love, patience, and everyday support made all of this possible. ♡

Vít Musil
March 2026

Author's Contribution

The habilitation process requires estimating author's contribution to individual papers included in the collection. Since precise percentages are impossible to determine, we use the proxy measure that assigns equal credit to all coauthors of a paper. Each paper's summary also includes a sketch of the author's actual contributions.

Declaration

I used generative AI tools during the preparation of the thesis for grammar checking, drafting article summaries, and polishing the text. These tools were used in accordance with the principles of academic integrity, and I take full responsibility for the final content of the thesis.

Contents

1 Introduction

Artificial intelligence is increasingly evaluated not only by predictive accuracy, but by its ability to support correct decisions under explicit structural constraints. This shift exposes a methodological fault line: statistical learning is formulated in continuous, differentiable spaces, whereas many decisions are discrete, combinatorial, and subject to exact feasibility constraints. The presented work addresses this issue from an optimization perspective by developing mathematically explicit interfaces between continuous training dynamics and discrete algorithmic execution, with emphasis on neuro-symbolic integration and differentiable programming methods that remain both computationally tractable and operationally faithful.

1.1 The Paradigm Clash: Continuous Learning vs. Discrete Decision Making

Modern artificial intelligence is built upon a fundamental dichotomy between continuous and discrete optimization. The dominant paradigm of machine learning, particularly deep learning, relies heavily on continuous, differentiable functions. Neural networks learn by mapping continuous inputs to continuous outputs, guided by gradient descent algorithms that iteratively adjust parameters to minimize a loss function. This requires the entire computational pipeline to be at least differentiable.

However, the real world is inherently discrete. Core computational tasks, traditionally under the hood of artificial intelligence, such as routing, resource allocation, and logical reasoning, are governed by combinatorial structures. When an autonomous system must make a hard decision, such as finding the shortest path or executing a security patrol action, it moves from the continuous domain into a discrete state space.

This creates a structural bottleneck in algorithmic design. If a continuous learning model incorporates a discrete algorithm (e.g., an integer programming module or a graph-matching algorithm), standard automatic differentiation fails. The derivative of a step function or a discrete $\arg\min$ operation is zero almost everywhere and undefined at the decision boundaries. Consequently, the gradient cannot flow backward through this algorithm, making end-to-end optimization impossible.

A symmetric challenge exists in game theory and agent planning. Synthesizing optimal strategies in stochastic or adversarial environments typically requires navigating a combinatorial explosion of discrete states. While continuous optimization methods hold the promise of finding efficient solutions in large state spaces, applying them to synthesize discrete, finite-memory strategies is not straightforward.

1.2 Objectives and Contributions

The central objective of our work is to bridge the gap between continuous gradient-based optimization and discrete algorithmic execution. Rather than treating machine learning and discrete planning as isolated pipelines, we present a unified mathematical and computational framework that enables complex AI systems to optimize objectives across discrete, combinatorial landscapes. Concretely, we pursue three tightly connected objectives:

- Develop principled gradient surrogates for non-differentiable operators, so that black-box combinatorial procedures can be embedded into end-to-end learning pipelines without sacrificing optimization stability, performance, and guarantees.

- Formulate systematic relaxation-and-projection schemes that translate discrete planning and finite-memory synthesis problems into tractable continuous programs while preserving decision-relevant structure.
- Validate these methods on realistic tasks where prediction quality alone is insufficient, and where final performance is determined by the quality of downstream decisions under constraints.

Taken together, our line of work advances a unifying perspective: discrete algorithmic reasoning and continuous statistical learning should not be connected by ad hoc interfaces, but by explicit optimization principles that preserve both differentiability for training and combinatorial validity at execution time.

1.3 Application Domains

The developed methods are applied and validated across two domains of computer science, where the continuous-discrete clash is especially pronounced. These form the two application pillars of this thesis:

Combinatorial Optimization (Machine Learning and Control). We address the structural limitations of modern machine learning architectures and differentiable simulators. Standard neural networks struggle to enforce hard logical constraints or execute exact algorithmic reasoning. We demonstrate how to natively embed non-differentiable operations, such as black-box combinatorial solvers, integer programming modules, and discontinuous physical contacts, directly within continuous learning loops. By resolving the vanishing or undefined gradients, we enable the end-to-end optimization of highly complex tasks, including graph matching, rank-based metric optimization, and sim-to-real robotic control.

Strategy Synthesis in Stochastic Environments (Game Theory and Planning). The second pillar shifts the focus from embedding algorithmic nodes to navigating discrete, combinatorial state spaces. In adversarial and stochastic environments, such as patrolling games, synthesizing optimal strategies is prone to state-space explosion. We demonstrate how the same philosophy of continuous optimization can be applied to discrete planning. By systematically relaxing discrete decisions into a space of probabilistic strategies and infinite-horizon objectives into differentiable formulations, we introduce scalable methods to compute resilient strategies that outperform classical discrete search techniques.

2 Differentiable Programming for Combinatorial Algorithms

The integration of deep learning with algorithmic reasoning represents one of the significant shifts in contemporary artificial intelligence. As we move beyond simple pattern recognition, the challenge lies in creating systems that can not only perceive but also reason and act optimally within structured environments.

From Perception to Deliberation

The landscape of artificial intelligence is characterized by a fundamental dichotomy, often framed through the lens of Daniel Kahneman's "System 1" and "System 2" cognitive processes. System 1, embodied by modern deep learning, excels at fast, intuitive, and high-dimensional pattern matching. These models are inherently statistical, mapping inputs x to predictions $\hat{y}$ through differentiable mappings $f_\theta(x)$. Conversely, System 2 represents the deliberative, procedural rigor of classical Operations Research (OR) and symbolic logic. These systems operate through explicit algorithmic reasoning, such as graph algorithms or Mixed-Integer Linear Programming, to find an optimal decision z^* that minimizes a cost function subject to rigid constraints.

While deep learning provides unprecedented predictive power, it often lacks the structural guarantees and logical consistency required for high-stakes decision-making. Traditional OR, while rigorous, typically assumes that the parameters of the optimization problem (e.g., travel times, consumer demand, or physical coefficients) are known a priori. In reality, these parameters are often unknown and must be estimated from noisy data. The central thesis of this research is that the frontier of AI lies in the seamless synthesis of these two paradigms: moving from "AI as a black-box predictor" to "Objective-Driven AI."

From the vantage point of 2026, foundation and large language models have made this boundary more porous, yet their most dependable use in high-stakes settings still relies on explicit optimization, constraint handling, and verifiable planning.

The Imperative for Objective-Driven AI

In industrial and scientific applications, the decoupling of prediction and optimization often leads to suboptimal outcomes. A standard "predict-then-optimize" pipeline minimizes a surrogate statistical loss, such as Mean Squared Error on the predictions $\mathcal{L}(y, \widehat{y}) = \|y - \widehat{y}\|^2$. However, in complex systems, a small predictive error in a critical parameter can lead to a catastrophically poor decision $z^*(\widehat{y})$, while a large error in a non-critical parameter might have no impact on the final utility.

On the other hand, purely predictive models frequently fail to respect hard physical laws or combinatorial constraints, producing "optimal" solutions that are infeasible in the real world. This mismatch motivates a tighter integration in which prediction quality is evaluated by its impact on the decisions made.

By constructing end-to-end trainable pipelines, we allow the model to understand the downstream task. The learning process is thus guided by the decision loss $\mathcal{L}(z, z^*(\widehat{y}))$, ensuring that the neural network learns to extract features that are most relevant to the ultimate objective of the system.

The Technical Bottleneck: Differentiating the Non-Differentiable

The primary obstacle to realizing this synthesis is the "vanishing gradient" problem inherent in discrete settings. Standard backpropagation relies on the Chain Rule to propagate errors from

the loss function back to the model parameters θ as

$$\nabla_\theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial z^*} \cdot \frac{\partial z^*}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta}. \tag{1}$$

When the component $z^*(\hat{y})$ is a combinatorial solver, the mapping from parameters to optimal solutions is a piecewise constant step function. Consequently, the derivative $\partial z^*/\partial \hat{y}$ is zero almost everywhere and undefined at the points of discontinuity.

Standard gradient-free approaches, including REINFORCE-style stochastic estimators and other zero-order optimization schemes, frequently exhibit variance and sample complexity that scale poorly with problem dimension. In practical settings where each objective evaluation requires solving an expensive combinatorial program (e.g., an ILP), this results in intractably slow training dynamics and prohibitive computational costs.

Another branch of methods replaces or softens the entire solver by fully differentiable surrogates, thereby enabling gradients to flow. While this strategy can improve optimization convenience, it typically weakens the key advantages of the original algorithmic component: exact feasibility, optimality guarantees, and the empirical performance delivered by decades of specialized solver engineering. Our objective is therefore different: to incorporate discrete algorithms in their native form and still enable end-to-end learning, without sacrificing the correctness, robustness, and efficiency properties that make these solvers valuable in the first place.

Overview of Contributions

This chapter is organized around two complementary contribution streams, each discussed in detail in the subsequent sections.

Methodological Foundations. This line of work established practical differentiation through black-box combinatorial solvers by using surrogate gradients. The resulting Blackbox-Backprop method [1] enables end-to-end training for decision quality in settings with arg min/arg max operators with linear costs. Next, we extended black-box differentiation to richer optimization layers and more expressive formulations. CombOptNet [2] learns integer-program structure directly from data, enabling models to recover problem formulations rather than only their parameters. Finally, LPGD [3] unifies several surrogate-gradient principles into a general mathematical framework and links them to the traditional optimization techniques.

Applications in Computer Vision and Physics Simulation. We demonstrated the applicability and effectiveness of our Blackbox Differentiation approach for rank-based metric optimization [4] and deep graph matching [5], achieving consistent gains across structured prediction tasks. We extended the applicability beyond combinatorial optimization to contact-rich physical simulation [6]. Hard-contact dynamics are non-smooth and therefore impede standard automatic differentiation; we address this by constructing soft gradient surrogates that preserve accurate forward simulation while yielding stable backward signals for learning and control.

2.1 Methodological Foundations

2.1.1 Differentiating through Black-Box Solvers

Our first contribution to this area was the Blackbox Differentiation paper accepted as a spotlight talk at ICLR 2020.

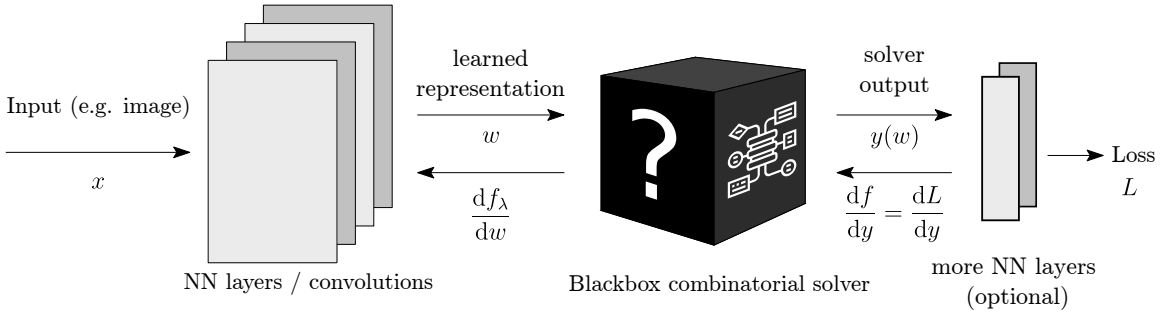


Figure 1: Architecture design enabled by our method: Blackbox solver embedded into a neural network, fully end-to-end trainable.

[1] Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. "Differentiation of Blackbox Combinatorial Solvers". In: *The Eighth International Conference on Learning Representations*. ICLR 2020. May 2020. URL: https://openreview.net/forum?id=BkevoJSYPB.

We developed a method for differentiating through black-box combinatorial solvers, enabling end-to-end training of models that incorporate discrete optimization components without requiring access to the internal structure of the solver as sketched in Fig. 1. The only assumption on the solver is that it solves an optimization task with *linear* objective over *any* feasible set $\mathcal{Y} \subset \mathbb{R}^m$. Specifically, given a cost $w \in \mathbb{R}^m$, the solver returns $y(w)$ s.t.

$$y(w) = \arg\min_{y \in \mathcal{Y}} \langle w, y \rangle. \tag{2}$$

There are no assumptions on the structure of the output space $\mathcal{Y}$; it need not be available or known to the method user. The most difficult situation occurs when $\mathcal{Y}$ is discrete, since $w \mapsto y(w)$ is piecewise constant (small changes to the costs do not change the optimal solution).

The goal is therefore to construct an informative backward signal $\nabla_w \mathcal{L}$. Our method resolves this by replacing the zero exact derivative with the gradient of a rigorously defined affine interpolation f_λ of the task loss around the current solver output. However, this interpolation is never explicitly constructed, which makes the method extremely efficient. All we need is just another solver call on the backward pass, see Algorithm 1. Operationally, this means one

Algorithm 1 Forward and backward passes of black-box backpropagation.

function FORWARDPASS(w)
  $y \leftarrow$ **Solver**(w) ▷ $y = y(w)$
  Store (w, y) for the backward pass
  return y

function BACKWARDPASS($\nabla_y \mathcal{L}(y), \lambda$)
  Load (w, y) from the forward pass
  $w' \leftarrow w + \lambda \nabla_y \mathcal{L}(y)$
    ▷ Calculate perturbed weights
  $y_\lambda \leftarrow$ **Solver**(w')
  return $\nabla_w f_\lambda(w) \leftarrow -\frac{1}{\lambda}(y - y_\lambda)$
    ▷ Gradient of continuous interpolation

standard solver call in the forward pass and one standard solver call in the backward pass with perturbed costs. No solver internals, KKT system, relaxation, or unrolling are required; any exact or heuristic combinatorial solver can be wrapped as long as it exposes the linear objective.

The key theoretical step is to define an auxiliary perturbed optimization problem $y_\lambda(w) = \arg\min_{y \in \mathcal{Y}} \{\langle w, y \rangle + \lambda f(y)\}$, where f is the local linearization of the outer loss $\mathcal{L}$ at y. Thanks

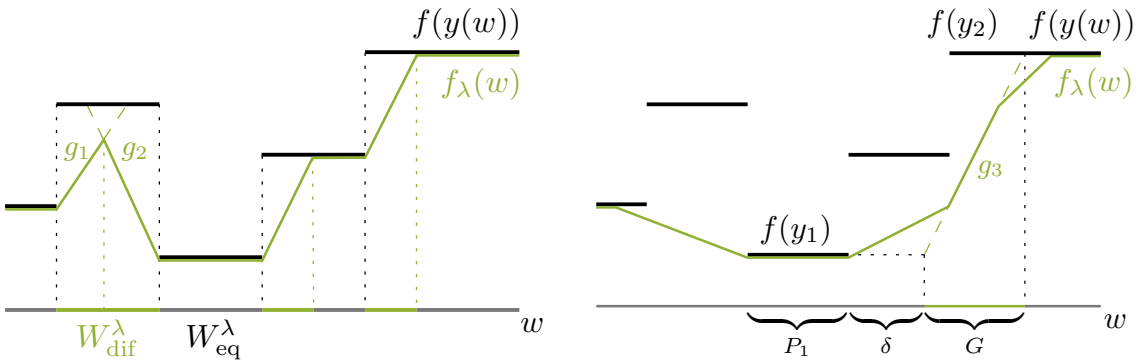


Figure 2: Continuous, piecewise affine interpolation f_λ (green) of a piecewise constant function $f(y(w))$ (black). For small values of λ, the interpolation is more faithful to the original function (left), whereas for large λ we get larger areas with informative derivatives (right).

to linearity, this can be solved by the given forward solver. From this, we construct a continuous piecewise-affine interpolation f_λ of the discrete objective and prove that

$$\nabla_w f_\lambda(w) = -\frac{1}{\lambda}(y(w) - y_\lambda(w)), \tag{3}$$

which is exactly the backward pass. Hence, the method is not an ad hoc straight-through estimator: it is the exact gradient of a well-defined smoothed surrogate objective induced by the combinatorial argmin structure.

The guarantees established in the paper are threefold: (i) f_λ is continuous and piecewise affine; (ii) the interpolation fidelity–smoothness trade-off is explicitly controlled by λ; and (iii) the resulting gradient captures neighboring decision-boundary changes that are invisible to the true zero Jacobian, yielding informative descent directions in practice.

Applicability is broad whenever the combinatorial layer is expressible via linear costs, such as shortest path, minimum-cost perfect matching, integer programming layers, and related structured prediction modules. In our paper, we demonstrated the applicability and scalability of the method on three synthetic problems incorporating the shortest path, the Traveling Salesman Problem, and min-cost perfect matching.

My contributions. I personally ensured the mathematical soundness of the methodology and helped maintain a careful balance between practical usefulness and formal correctness. I wrote the method section, theorems, and proofs, contributed to the conceptualization of the paper, and supported the writing, typesetting, and figure preparation. (20%)

2.1.2 Learning the Entire Combinatorial Module

With the current framework, a hybrid architecture can be built around an algorithm or a solver for a concrete combinatorial problem. While this is sufficient in some cases, much greater expressive power lies in the ability to jointly learn the problem specification. We introduced CombOptNet, which was accepted at ICLR 2021.

[2] Anselm Paulus, Michal Rolínek, Vít Musil, Brandon Amos, and Georg Martius. "CombOptNet: Fit the Right NP-Hard Problem by Learning Integer Programming Constraints".

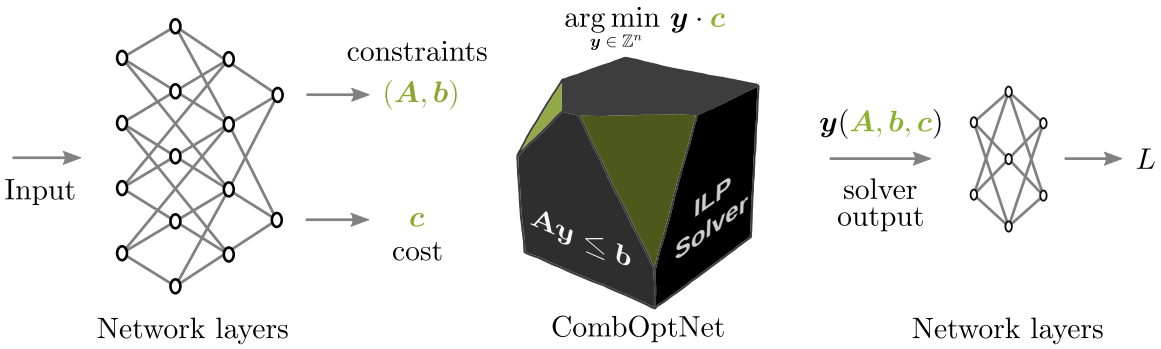


Figure 3: CombOptNet as a module in a deep architecture. It enables learning the entire combinatorial specification, including both the cost vector and the constraints, from data.

In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 8443–8453. URL: https://proceedings.mlr.press/v139/paulus21a.html.

CombOptNet addresses a central limitation of early decision-focused learning architectures: they typically learn only the objective coefficients of a *fixed* combinatorial problem, while the structural constraints are assumed known in advance. The paper removes this assumption and proposes an end-to-end trainable integer-programming layer that can jointly infer (i) an instance-dependent cost vector and (ii) the combinatorial specification itself, represented by linear inequality constraints. This is a major conceptual step from "predicting scores for a predefined solver" to "learning which combinatorial task is solved."

Formally, the layer solves

$$y(A, b, c) = \arg\min_{y \in \mathcal{Y}} \langle c, y \rangle \quad \text{subject to} \quad Ay \leq b, \tag{4}$$

where $\mathcal{Y} \subseteq \mathbb{Z}^n$ is bounded, where the neural parameters θ are optimized from task supervision and both (A, b, c) are learnable. See Fig. 3 for an overview. The technical challenge is that changing constraints cause discontinuous jumps of the feasible polytope and of the optimal vertex, so direct differentiation is unavailable. CombOptNet resolves this by combining exact integer optimization in the forward pass with a principled surrogate-gradient design in the backward pass, based on local geometry around the active solution and a decomposition in a suitable basis of perturbation directions. This preserves solver exactness at inference while providing informative training signals.

An important innovation is the parameterization of constraints. Instead of learning raw half-space coefficients only, the method introduces a representation through learnable normals, offsets, and geometric reparameterizations (origin–distance style) that make optimization over constraint sets substantially better conditioned. Empirically, this avoids unstable updates that would otherwise correspond to large rotations/translations of constraints in coefficient space.

On synthetic tasks (random-constraint families and weighted set covering), the model recovers multi-constraint combinatorial structure from supervision. The key finding is that linear-program relaxation surrogates fail when the decision boundary is intrinsically integral, while CombOptNet succeeds by differentiating through exact IP solutions. In a language-conditioned KNAPSACK task, input-dependent constraints and objectives are learned jointly from embeddings, demonstrating that learning the discrete specification improves performance over fixed-structure baselines.

The main large-scale demonstration is keypoint matching on SPair-71k. Here, the architecture intentionally omits hand-crafted matching constraints and infers them from data. Despite this more challenging setting, CombOptNet achieves strong accuracy across problem sizes. This provides quantitative support for the paper's narrative: flexible learning of the combinatorial specification can approach specialized architectures while offering broader applicability.

My contributions. I contributed to the paper's conceptualization and to the method design. I wrote the method section, including the central claims and proofs. I further contributed to the overall manuscript writing and to the preparation of graphics. (20%)

2.1.3 LPGD: A Unified Framework

The following work provides a broader mathematical framework for discrete optimization layers and unifies a collection of seemingly different surrogate-gradient tricks into a single principle. The paper was accepted at ICML 2024.

[3] A. Paulus, G. Martius, and V. Musil. "LPGD: A General Framework for Backpropagation through Embedded Optimization Layers". In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. Proceedings of Machine Learning Research. PMLR, July 2024, pp. 39989–40014. URL: https://proceedings.mlr.press/v235/paulus24a.html.

The work is devoted to learning pipelines in which a network predicts parameters w of an embedded optimization primal-dual problem of the general form

$$z^*(w) = (x^*(w), y^*(w)) \in \arg\min_{x \in X} \max_{y \in Y} \mathcal{L}(x, y, w), \tag{5}$$

while training minimizes an outer loss $\ell(x^*(w))$. Our approach is based on a *Lagrange–Moreau envelope*. Instead of measuring proximity to the current solution by Euclidean distance alone, we introduce the Lagrangian divergence

$$D_{\mathcal{L}}(x, y \mid w) = \mathcal{L}(x, y, w) - \mathcal{L}^*(w), \tag{6}$$

which quantifies how far (x, y) is from optimality for the current parameters. Combining the task loss with the scaled divergence $D_{\mathcal{L}}/\tau$ yields lower, upper, and averaged smoothed objectives, with $\tau > 0$ controlling an explicit tightness–smoothness trade-off. Gradient descent on these envelopes defines the *Lagrangian Proximal Point Method*.

Due to the loss term, efficiently solving the backward problem may require a custom algorithm slower than the forward oracle. Instead, we aim to introduce a justifiable approximation that allows the computation with the same forward solver oracle. In typical cases, the Lagrangian contains a linear term and decomposes as

$$\mathcal{L}(x, y, w) = \langle x, c \rangle + \Omega(x, y, v), \quad \text{with} \quad w = (c, v). \tag{7}$$

Using this, the backward pass requires only a perturbation of the linear parameter block

$$\tilde{z}_\tau(w) = z^*(c + \tau \nabla \ell(x^*), v), \qquad \nabla_w \tilde{\ell}_\tau(w) = \frac{1}{\tau}(\nabla_w \mathcal{L}(\tilde{z}_\tau, w) - \nabla_w \mathcal{L}(z^*, w)). \tag{8}$$

This is the LPGD update. Operationally, this means the exact forward solver remains unchanged, and the backward pass is obtained via a single additional oracle call on perturbed

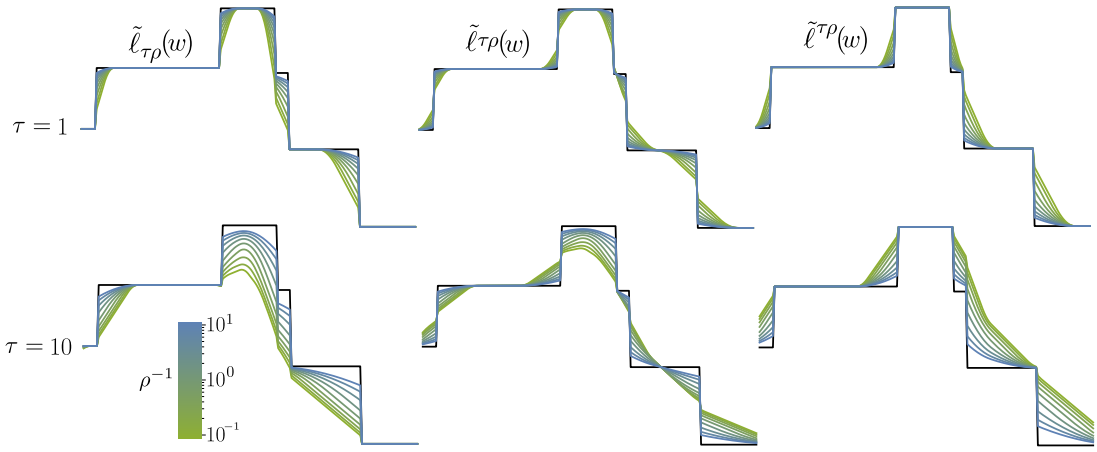


Figure 4: LPGD surrogate loss landscape. The original loss is piecewise constant, while the upper, average, and lower Lagrange-Moreau envelopes are piecewise differentiable, with informative gradients that capture changes near the decision boundary.

parameters, typically warm-started from the forward solution. The paper further introduces an optional quadratic regularization parameter ρ that augments the envelope around the current primal solution, connecting it to classical proximal and projection-based updates. An example of the resulting surrogate loss landscape is shown in Fig. 4.

Several methods that previously appeared unrelated emerge as special cases or asymptotic limits of a single envelope-based framework. For linear programs, the construction collapses to the Blackbox Backpropagation rule. In the limit $\tau \to 0$, LPGD recovers the true gradient whenever a differentiable selection of the solution mapping exists, thereby connecting the method to implicit differentiation and finite-difference perturbation schemes. In the opposite regime, $\tau \to \infty$, the updates converge to proximal, Frank-Wolfe, or projection-type steps on the effective feasible set, thereby recovering the principles underlying Identity with Projection, SPO+, and Fenchel-Young losses. The paper further proves that the envelopes become smoother as τ increases, formalized by explicit Lipschitz bounds, and it analyzes the effect of inexact solver calls. Conceptually, it again recasts surrogate gradients not as heuristic replacements for missing derivatives, but as exact gradients of carefully chosen smoothed objectives.

The empirical study demonstrates the framework's utility beyond settings with degenerate derivatives through two experiments: learning Sudoku rules from puzzle pairs and tuning Markowitz portfolio policies. In both cases, LPGD outperforms standard gradient descent, reaching lower errors, faster convergence, and better objective values.

My contributions. I supervised the project, contributed to the paper's conceptualization, method design, and mathematical framing. I contributed to formulating the statements and proofs, and to writing the paper. (33%)

2.2 Applications in Computer Vision and Physics Simulation

2.2.1 Direct Rank-based Metrics Optimization

Upon introducing Blackbox Differentiation, we set out to demonstrate its applicability beyond synthetic benchmarks. We showed that we can directly optimize relevant non-decomposable

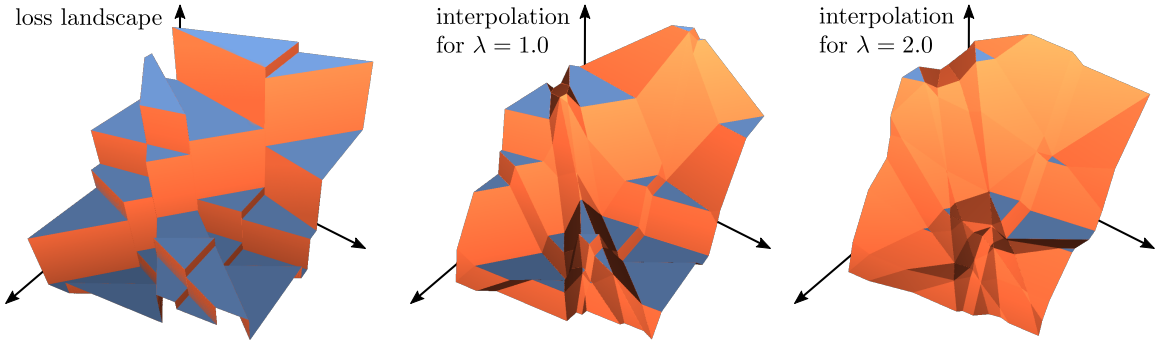


Figure 5: Differentiation of a piecewise constant rank-based loss. A two-dimensional section of the loss landscape is shown (left) along with two efficiently differentiable interpolations of increasing strengths (middle and right).

ranking objectives in modern vision pipelines. This resulted in an oral presentation at CVPR 2020.

[4] Michal Rolínek, Vít Musil, Anselm Paulus, Marin Vlastelica, Claudio Michaelis, and Georg Martius. "Optimizing Rank-based Metrics with Blackbox Differentiation". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Best paper nomination. Piscataway, NJ: IEEE, June 2020, pp. 7617–7627. DOI: 10.1109/CVPR42600.2020.00764.

This paper addresses an optimization gap between *what we evaluate* and *what we train* in visual recognition. In retrieval and detection, model quality is typically reported using rank-based criteria such as Recall@K and Average Precision (AP), yet training is often performed with proxy losses (pairwise, triplet, cross-entropy, or smooth surrogates) that do not align with these metrics. The central technical obstacle is that ranking metrics are non-differentiable and non-decomposable over examples, because they depend on discrete sorting decisions and global interactions across the ranked list.

The key contribution is to cast ranking as a black-box combinatorial layer and differentiate it with the Blackbox Differentiation principle. Concretely, for a score vector $y \in \mathbb{R}^n$, ranking is represented as a linear-cost combinatorial optimization problem

$$\mathrm{rk}(y) = \operatorname*{arg\,min}_{\pi \in \Pi_n} \langle y, \pi \rangle, \tag{9}$$

where Π_n is the set of all permutations of $\{1, \ldots, n\}$. Hence, minimizing (9) returns the permutation that assigns smaller rank indices to larger scores. Instead of replacing this exact discrete solver with a relaxation, the method keeps the *exact* ranking operator on the forward pass and obtains an informative backward signal via one additional call on perturbed costs. In practical terms, this yields a mini-batch loss whose gradient is consistent with a mathematically defined smoothed objective and is computable with standard optimizers. The visualization of the implicit linearly interpolated loss landscape is shown in Fig. 5.

Two further innovations are essential for stable large-scale training. First, the paper introduces *Score memory*: instead of ranking only within the current mini-batch, it maintains a memory of scores from recent iterations and uses this enlarged candidate pool when constructing rank-based losses. This approximates full-dataset ranking, reduces mini-batch bias and variance,

and provides a substantially denser signal for each update. Second, it introduces a *Score margin* design: positive examples are required to exceed competing negatives by an explicit score gap, which regularizes near-tie regions and suppresses unstable rank oscillations. The margin therefore stabilizes AP-oriented optimization by turning small, noisy score differences into controlled separations aligned with ranking semantics. Third, the *Recall loss* is designed directly at the rank level and can be written as an equivalent weighted sum of recall-at-k over all cutoffs k,

$$\mathcal{L}_{\text{rec}} = \sum_{k=1}^{\infty} w_k \, \ell_k, \qquad \ell_k \equiv 1 - \text{Recall@}k, \tag{10}$$

with nonnegative weights w_k specifying which parts of the ranked list are emphasized. Different weighting schemes yield distinct training criteria (e.g., stronger top-of-list emphasis versus flatter global recall pressure), forming a unified family of recall-oriented objectives. Within this view, several earlier ad hoc formulations that were only reported as empirically "beneficial" are recovered as specific choices of weights which provide their missing theoretical explanation.

We evaluate the method in two canonical domains where rank metrics are first-class objectives. (i) *Image retrieval:* the method is trained directly for rank-based retrieval criteria and compared against strong metric-learning baselines. Across standard retrieval benchmarks, it reaches performance competitive with state-of-the-art methods while using a conceptually cleaner objective. (ii) *Object detection:* AP-driven black-box training is integrated into near state-of-the-art detector pipelines. Here, the method consistently improves detection quality relative to the corresponding baseline training objectives, demonstrating that the approach is not limited to small, controlled ranking tasks but scales to modern dense prediction systems.

The principal empirical message is robust: direct optimization of rank-based targets is both feasible and beneficial when coupled with principled black-box differentiation. In retrieval, one obtains top-tier accuracy without bespoke task-specific relaxations. In detection, one observes consistent gains from better alignment between the training and test objectives.

My contributions. I contributed to the conceptual formulation linking rank-metric optimization to blackbox differentiation and co-designed the objective construction. I wrote the method section, core claims and proofs, and contributed to the implementation and visualizations. (16%)

2.2.2 Applications to Graph Matching

Our next goal was to demonstrate the superiority of Blackbox Differentiation on a combinatorially complex problem. This resulted in a paper accepted at ECCV 2020.

[5] Michal Rolínek, Paul Swoboda, Dominik Zietlow, Anselm Paulus, Vít Musil, and Georg Martius. "Deep Graph Matching via Blackbox Differentiation of Combinatorial Solvers". In: *Computer Vision – ECCV 2020*. Vol. 28. Lecture Notes in Computer Science, 12373. Cham: Springer, Aug. 2020, pp. 407–424. DOI: 10.1007/978-3-030-58604-1_25.

This work is devoted to end-to-end learning of semantic keypoint correspondence under an exact combinatorial-matching objective. Given two images with detected keypoints, the task is to infer a partial one-to-one mapping that is simultaneously appearance-consistent and geometrically coherent; see Fig. 6.

Formally, for graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, one optimizes over admissible matchings $(v, e) \in \text{Adm}(G_1, G_2)$, where $v \in \{0,1\}^{|V_1||V_2|}$ encodes matched vertices and $e \in \{0,1\}^{|E_1||E_2|}$

Figure 6: Given two images with annotated keypoints, the task is to match keypoints with the same semantics in both images (e.g., left wing, nose). Example matchings from the SPair-71K dataset.

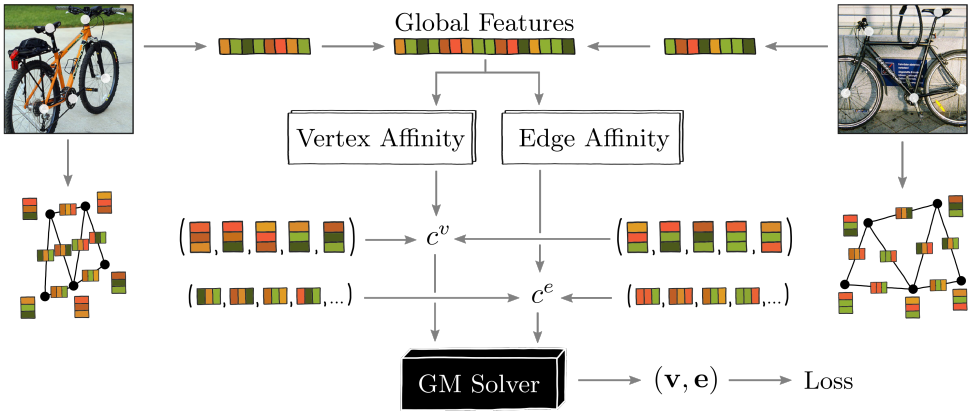


Figure 7: Construction of a combinatorial instance for keypoint matching.

induced edge correspondences. With learned unary and pairwise costs (c_v, c_e), the combinatorial core is the assignment-type optimization problem

$$\min_{(v,e)\in \mathrm{Adm}(G_1,G_2)} \langle c_v, v\rangle + \langle c_e, e\rangle. \tag{11}$$

This formulation ensures we can apply Blackbox Differentiation regardless of which solver we choose for this task in practice. Only what matters is that the objective function in optimization 11 is linear. The structure of the set $\mathrm{Adm}(G_1, G_2)$ is not relevant.

Following the Blackbox Differentiation principle, we use an *unmodified* high-performance combinatorial solver on the forward pass, while the backward pass computes an informative surrogate gradient through one additional solver call on the perturbed inputs. Hence, the method preserves the algorithmic strength of exact/discrete optimization while remaining trainable within standard first-order deep learning loops. For reference, we call it BB-GM (Blackbox Backprop Graph Matching).

Architecturally, BB-GM combines three components: (i) CNN feature extraction at keypoints together with a global image descriptor; (ii) geometry-aware refinement via graph neural processing (SplineCNN-style message passing over keypoint graphs); and (iii) construction of combinatorial costs passed to a state-of-the-art graph matching solver based on Lagrangian decomposition and dual block-coordinate ascent. An important conceptual novelty is the global-feature attention mechanism, which reweights node/edge affinities based on scene-level context (viewpoint, scale, rigidity), improving disambiguation under occlusion and repeated local patterns.

The experimental protocol is broad and methodologically careful: on Pascal VOC (Berkeley

keypoints) in the standard intersection-filtered setting, BB-GM achieves 80.1 ± 0.6 mean matching accuracy, substantially outperforming reproduced DGMC* (73.2 ± 0.5) and earlier baselines CIE (68.9), GLMNet (67.5), and NGM+ (66.1); in the more realistic *unfiltered* setting with outliers and unequal keypoint counts, BB-GM reaches 61.4 ± 0.5 F1, improves to 62.8 ± 0.5 with multi-graph cycle-consistency post-processing, and clearly exceeds a forced-maximal-matching ablation (51.9 ± 1.0), confirming the value of native partial matching; on SPair-71k, the method again improves over DGMC* from 72.2 ± 0.2 to 78.9 ± 0.4 overall, with strong gains across viewpoint difficulty bins (easy: 79.4→84.8, medium: 65.2→73.1, hard: 61.3→70.6), while Willow ObjectClass results remain competitive across pretraining/fine-tuning protocols.

From a methodological perspective, the paper demonstrates that one does not need to weaken combinatorial inference into a soft surrogate to obtain trainability. Instead, a strict separation of concerns is achieved: neural components learn cost structure from data, while the solver enforces global discrete consistency. This has two long-term implications highly relevant for neuro-symbolic AI: (a) architectural modularity, since stronger solvers can be swapped in without redesigning the learning backbone; and (b) distributional robustness in deployment settings that differ from training-time filtering assumptions.

My contributions. I co-designed the blackbox-differentiation-based integration of the combinatorial solver into the deep graph-matching pipeline and shaped the method's mathematical and algorithmic framing. I wrote the method section, contributed to the experimental design (including the more challenging unfiltered evaluation protocol), manuscript writing, result presentation, and figure drawing. (16%)

2.2.3 Differentiable Physics Simulation

Finally, we tackled the problem of differentiable physics simulation in the following ICLR 2026 paper.

[6] Anselm Paulus, Andreas René Geist, Pierre Schumacher, Vít Musil, Simon Rappenecker, and Georg Martius. "Differentiable Simulation of Hard Contacts with Soft Gradients for Learning and Control". In: *The Fourteenth International Conference on Learning Representations*. ICLR 2026. 2026. URL: https://openreview.net/forum?id=2EGtfFwxx8.

This work addresses one of the main obstacles in differentiable robotics and control: physically realistic contact dynamics are intrinsically hard and discontinuous, whereas gradient-based learning requires derivatives that are both stable and informative. In simulators such as MuJoCo, one can obtain gradients by softening contacts, but this comes at the price of distorted dynamics and a larger sim-to-real gap. Conversely, if the simulator is configured for hard, realistic contacts, automatic differentiation through the resulting stiff dynamics produces gradients that are highly inaccurate or entirely useless. Formally, we work with discrete-time dynamics

$$x_{k+1} = \mathrm{step}(x_k, a_k, p),$$

where x_k is the simulator state, a_k the control action, and p the physical parameters, and the task is to differentiate a long-horizon loss through collision-rich trajectories.

Our first contribution is a diagnosis of why gradients fail in penalty-based contact simulation. We showed that the dominant source of error is not merely a non-smoothness of the objective, but the numerical integration of stiff contact ODEs with a fixed timestep. Based on this observation, we introduce *DiffMJX*, which combines MuJoCo XLA with adaptive-step

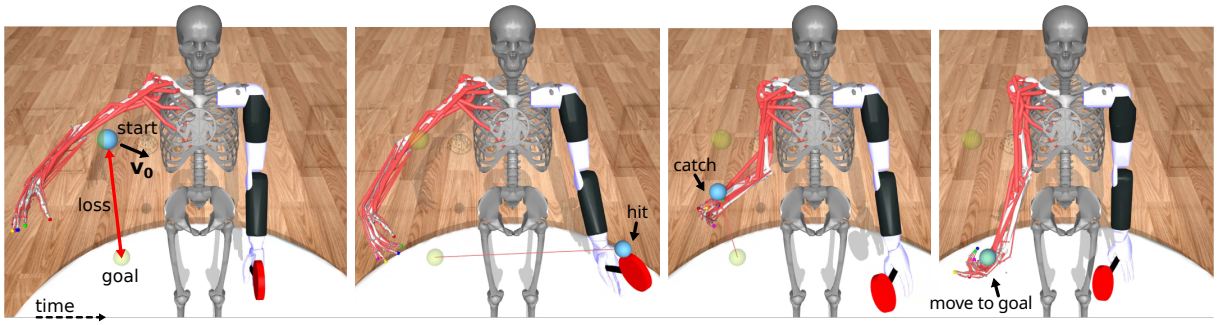


Figure 8: Autodiff-driven model predictive control with *Contacts from Distance* on the bionic tennis task. Task completion requires the racket to deflect the ball towards the MyoArm, which uses 63 muscle-tendon actuators to catch the ball and move it to the goal position. Only the distance between the ball and the goal is used as the cost.

integration from Diffrax and additionally smooths several low-level discontinuities in collision detection. The benefit is that the numerical precision is increased only near collision events, where stiffness is large, rather than by globally shrinking the timestep everywhere, which would be computationally prohibitive.

The second contribution targets the opposite failure mode: the vanishing of gradients before contact. If two bodies are still separated, ordinary simulator gradients are exactly zero with respect to motions that would bring them into contact later. This makes gradient-based planning fundamentally myopic. To overcome this, we propose *Contacts from Distance (CFD)*: the contact impedance is smoothly extended to a small band of positive-signed distances, allowing the backward model to generate weak virtual pre-contact forces. The crucial point is that these artificial forces are used only in the backward pass. Hence, the forward trajectory remains physically faithful, while the backward pass receives informative contact-seeking gradients.

Experiments span three settings: on toy collision systems, DiffMJX eliminates gradient oscillations and matches central differences by orders of magnitude. On ContactNets cube-toss, system identification achieves $\sim$5% error with CFD improving robustness to poor initialization; and in gradient-based MPC on MyoSuite, DiffMJX outperforms sampling-based baselines. CFD proves decisive for weakly supervised tasks, most notably bionic tennis, where vanilla gradients fail entirely, and CFD enables discovery of the full bounce–catch–transport sequence (see Fig. 8).

Overall, the paper shows that one does not have to choose between realistic hard-contact simulation and trainable gradients. Adaptive integration resolves the *correctness* problem of stiff-contact differentiation, and CFD resolves the *informativeness* problem of pre-contact optimization.

My contributions. I contributed to the mathematical framing of the problem, the conceptualization of the method section, and the writing of the paper. (16%)

3 Strategy Synthesis in Stochastic Environments

Strategy synthesis in stochastic environments occupies the same conceptual boundary as differentiable programming, but the interface is shifted. Here, the challenge is not to differentiate through a solver embedded in a neural architecture; it is to optimize over strategies that govern a stochastic or adversarial system over an unbounded horizon. The agent acts in a space of hard, atomic moves, while performance is defined by chance of detection, long-run average rewards, or resilience under changing conditions. The resulting synthesis task is therefore neither purely combinatorial nor inherently continuous. However, the unifying theme remains the same: to apply continuous optimization principles to solve combinatorial decision problems that are otherwise intractable at scale.

From Pure Actions to Randomized Control

Applications in security, maintenance, autonomous monitoring, and service logistics are intrinsically sequential. The performance criterion is rarely an instantaneous reward; rather, it is a long-run objective capturing reliability, robustness, frequency of visits, exposure to attacks, etc. Consequently, the object of optimization is a *strategy* over time.

In adversarial models, deterministic policies are often suboptimal. A pure strategy is easy to interpret but also predictable and exploitable by an adversary. Randomization is therefore not a secondary modeling choice, but a structural ingredient of optimal control. Hence, the strategies must assign probabilities to actions.

This is the first step in the continuous-discrete bridge: the decision variable is no longer a discrete action, but a *probability distribution* over actions. This turns combinatorial choices into continuous variables on probability simplices, enabling gradient-based updates.

Strategy Representation

In the considered models, pure or memoryless controllers are generally insufficient as they fail to realize the visitation frequencies, recurrent patterns, or local regularity constraints required by the objective. Therefore, we require randomized, history-aware behavior. To keep synthesis tractable, we compress history into finite *memory* and optimize a parametric finite-memory strategy.

Formally, we consider MDPs and patrolling games on graphs. Let S be states, A actions, and $P(s' \mid s, a)$ transition probabilities. A history h_t at (discrete) time t is a sequence of states and actions observed up to time t, i.e., $h_t = (s_0, a_0, s_1, a_1, \ldots, s_t)$, and a general (history-dependent) strategy is a map from histories to action distributions

$$\sigma(\cdot \mid h_t) \in \Delta(A). \tag{12}$$

A *finite-memory strategy* is specified by a finite memory set M, an initial memory distribution $\mu_0 \in \Delta(M)$, an action-selection map σ_a, and a memory-update map σ_m satisfying

$$\sigma_a(\cdot \mid s, m) \in \Delta(A), \quad \text{and} \quad \sigma_m(\cdot \mid s, m, a, s') \in \Delta(M), \tag{13}$$

with execution in the product system $(s, m) \in S \times M$. Intuitively, the controller observes (s, m), samples an action via σ_a, and then updates memory via σ_m based on the transition.

In a simplified graph-based environment, the system is modeled as a graph $G = (V, E)$, where vertices $V = S$ represent states, edges $E = A$ represent possible transitions, and $P(v' \mid v, e) = 1$

if and only if e is the edge from v to v'. In this case, the controller's state space is the product $V \times M$, and a finite-memory strategy is a map from $V \times M$ to distributions over target vertices and memory updates.

The memory state m is a compressed summary of history: instead of storing the full h_t, the controller stores only a statistic of the past that is relevant for future decisions. Two histories ending at the same vertex may differ combinatorially, but if they induce the same memory element m, the controller deliberately treats them as equivalent for future decisions.

Importantly, for key objective classes, we have ε-approximation guarantees: for every $\varepsilon > 0$, a finite-memory strategy exists within ε of the optimal history-dependent value.

Combinatorics of Memory and Randomization

Introducing memory is a double-edged sword: while it enables history-dependent behavior, it also lifts the closed-loop dynamics from $|S|$ states to the product space $|S||M|$. Many tasks encode complex long-run visitation patterns (frequencies, regularity, recurrence structure) for which optimal or near-optimal strategies may provably require large memory, making the design space both vast and structured. As a consequence, the synthesis objective for finite-memory controllers typically yields a highly nonconvex optimization landscape with many local optima.

Randomization mitigates this combinatorial burden. In adversarial settings, it prevents the opponent from exploiting predictability, as already discussed. However, it is equally useful in non-adversarial models as it can replace explicit counting by probabilistic mixing. For instance, consider a two-state system that must be visited with prescribed stationary frequencies: a deterministic finite-memory controller needs to keep track of visit counts to approximate them; in contrast, a memoryless randomized strategy can achieve the target frequencies by appropriately sampling actions.

Randomization, therefore, allows us to work with tractable memory sizes by achieving objective satisfaction *in expectation*, at the price of sample-path variability, a tradeoff that is acceptable when deviations are tolerable or can be bounded, but not when hard guarantees are required.

When deterministic execution is ultimately needed, we can post-process a randomized solution into a deterministic finite-memory controller (typically with increased memory) or project it onto a discrete strategy class. Overall, we cross the discrete–continuous bridge by optimizing over randomized finite-memory strategies on simplices, and, if necessary, return to a deterministic implementation at deployment.

Continuous Optimization of Finite-Memory Strategies

For a fixed memory architecture, a generic synthesis template is

$$\text{optimize } J(\sigma) \qquad \text{over } \sigma \in \Sigma_{\mathrm{FM}}(M) \qquad \text{subject to } C_k(\sigma) \le 0,\ k = 1, \dots, r, \tag{14}$$

where J encodes the task objective (e.g., detection, reachability, mean payoff) and C_k encode structural constraints. Crucially, J is a global functional of induced trajectories, not a local function of individual parameters. A perturbation at a single pair (v, m) propagates through the entire path distribution, alters recurrent classes and hitting behavior, and, in adversarial settings, modifies the opponent's best response. This nonlocal dependence is precisely why the evaluation map must be both expressive and computationally efficient.

We optimize a continuous relaxation of randomized finite-memory strategies, then recover executable controllers at deployment. Actions and memory updates are represented by local distri-

butions, enabling first-order constrained optimization. If necessary, sparse or discrete strategies are recovered by vertex convergence or by projection/sampling with feasibility restoration.

The critical technical bottleneck is the evaluation map. Before any improvement step is meaningful, one must define a strategy-evaluation function $\mathrm{Val}(\sigma)$ that is simultaneously (i) semantically faithful to the original objective J, (ii) computationally tractable at scale, and (iii) differentiable—or equipped with a reliable differentiable surrogate $\widetilde{\mathrm{Val}}(\sigma)$. Without this triad, optimization either becomes too slow for realistic instances, unstable due to noisy gradients, or misaligned with the true decision criterion. Smoothing is therefore not cosmetic but structural: when direct differentiation is unavailable, carefully constructed surrogates deliver informative gradients while preserving the practically relevant ordering of strategies.

At a high level, the workflow is as follows:

1. **Model and objective specification.** Fix the environment model (MDP or graph game), memory allocation, and a target objective.
2. **Continuous formulation.** Define a feasible continuous parameterization $\theta \mapsto \sigma_\theta$ (local action and memory probabilities), either directly on simplices or via unconstrained weights with softmax or projection, together with a strategy evaluation functional $\mathrm{Val}(\sigma)$.
3. **Differentiable evaluation design.** Compute $\mathrm{Val}(\sigma_\theta)$ exactly on the induced product process when possible; otherwise construct a differentiable proxy $\widetilde{\mathrm{Val}}(\sigma_\theta)$ with controlled approximation error and stable gradients.
4. **Optimization loop.** Initialize a random feasible strategy θ_0. Iterate evaluation, sensitivity computation (exact gradient or surrogate gradient), and update parameters by gradient descent, projected gradients, or related first-order methods.
5. **Discrete recovery and certification.** Return a deployable discrete/finite-memory strategy by direct vertex convergence, projection, or sampling, and certify its value in the original model.

Overview of Contributions

This chapter is structured around two contribution streams that jointly develop continuous optimization methods for finite-memory strategy synthesis in stochastic and adversarial environments.

Adversarial Patrolling Games. The first contribution stream develops scalable synthesis procedures for defender strategies in infinite-horizon adversarial patrolling games. Regstar [7] provides a tractable protection objective together with gradient computation for randomized finite-memory strategies on weighted graphs with imperfect detection. On-the-fly adaptation extends this viewpoint to non-stationary environments, where a previously deployed strategy must be updated quickly without opening security holes during the transition [8]. Finally, automatic memory assignment removes a major bottleneck by automatically deciding where controller memory should be allocated, making finite-memory patrolling usable without expert-designed state spaces [9].

Long-Run Objectives in Stochastic Environments. The second stream transfers the same optimization philosophy from patrolling games to broader long-run control criteria in stochastic systems. General recurrent reachability objectives [10] introduce an expressive optimization language built from hitting-time moments and edge frequencies, allowing one to

optimize not only performance but also stochastic stability. Mean-payoff optimization for periodic service and maintenance [11] then shows how randomized finite-memory controllers can compactly represent long-horizon behavior and how high-quality deterministic schedules can be recovered from them. Taken together, these works establish a common framework for evaluating and optimizing infinite-horizon objectives using continuous methods over randomized finite-memory strategies.

3.1 Adversarial Patrolling Games

3.1.1 Strategy Synthesis for Adversarial Patrolling Games

I personally entered the domain of patrolling games through a collaboration on the Regstar paper, which was accepted at UAI 2021.

[7] David Klaška, Antonín Kučera, Vít Musil, and Vojtěch Řehák. "Regstar: Efficient Strategy Synthesis for Adversarial Patrolling Games". In: *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*. PMLR, Dec. 2021, pp. 471–481. URL: https://proceedings.mlr.press/v161/klaska21a.html.

The work is devoted to infinite-horizon adversarial patrolling on directed graphs with arbitrary edge traversal times and possibly imperfect intrusion detection. Patrolling is modeled as a Stackelberg game: the Defender commits to a randomized patrol, and the Attacker chooses the target and launch time so as to maximize expected damage. Formally, a patrolling instance is given by $G = (V, T, E, \text{time}, d, \alpha, \beta)$, where $T \subseteq V$ is the set of targets, $d(\tau)$ is the attack duration and $\alpha(\tau)$ is the loss caused by a successful intrusion at target τ, and $\beta(\tau)$ is the detection probability upon a visit to τ. The objective is to maximize the guaranteed protection level against the Attacker's best response.

We work with finite-memory randomized (called *regular*) strategies and show that they approximate the optimal game value well, i.e.,

$$\sup_{\sigma \text{ is regular}} \text{Val}_G(\sigma) = \text{Val}_G,$$

where Val_G is the best achievable protection level in the original game, and $\text{Val}_G(\sigma)$ is the protection level guaranteed by a strategy σ. Next, we derive a tractable evaluation formula for a tight lower bound on $\text{Val}_G(\sigma)$, which is differentiable with respect to the local probabilities of σ, namely,

$$\text{Val}_G \geq \text{RVal}_G(\sigma) = \max_{\tau \in T} \alpha(\tau) - \max_{e,\tau} \{\alpha(\tau) - D(e, \tau \mid \sigma)\}, \tag{15}$$

in which $D(e, \tau \mid \sigma)$ aggregates, over all eligible continuations following an edge e, the expected damage from an attack on τ before $d(\tau)$ expires. The Attacker may select the weakest attack scenario, and hence $\text{RVal}_G(\sigma)$ is the corresponding guaranteed protection.

We then present a dynamical-programming algorithm for computing $D(e, \tau \mid \sigma)$ and its gradient. A naive treatment of $D(e, \tau \mid \sigma)$ would sum over exponentially many eligible paths that can reach τ within the attack horizon. Regstar avoids this blow-up by performing a reverse search from each target, aggregating all partial paths with the same endpoint, memory element, and elapsed traversal time. The method computes $D(e, \tau \mid \sigma)$ and $\nabla_\sigma D(e, \tau \mid \sigma)$ for all eligible edges simultaneously, with complexity polynomial in the size of the regular-strategy graph and in the number of distinct attainable traversal times.

Once this evaluator is available, synthesis is performed via gradient ascent, with normalization to valid local probability distributions. We also relax the hard maximum in (15) to provide denser gradient signals. In practice, Regstar is a multi-start local search method: it samples many random regular strategies, improves each with gradient-based updates, and returns the best one found. Methodologically, this is a clean use of differentiable programming in algorithmic game theory: gradients are used directly to optimize a finite-memory randomized controller.

Regstar is evaluated on synthetic benchmarks, a real Montreal ATM network, and office-building layouts. Across all settings, it matches or exceeds prior strategy-improvement quality while scaling better, solving instances where the baseline times out. Performance improves consistently with larger memory budgets, and in the office case, perfect protection was found for $|M| = 4$, showing that sufficient memory enables nontrivial patrol cycles.

Overall, the paper turns finite-memory patrolling synthesis from a largely combinatorial search problem into a differentiable optimization problem with formal guarantees and convincing empirical payoff.

My contributions. I contributed to the paper's conceptualization, optimization, evaluation design, and to its presentation and writing. (25%)

3.1.2 Strategy Synthesis in Changing Environments

In the follow-up work, we revisit our algorithm and extend it to the setting of changing environments, which was accepted at UAI 2022.

[8] Tomáš Brázdil, David Klaška, Antonín Kučera, Vít Musil, Petr Novotný, and Vojtěch Řehák. "On-the-Fly Adaptation of Patrolling Strategies in Changing Environments". In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. PMLR, Aug. 2022, pp. 244–254. URL: https://proceedings.mlr.press/v180/brazdil22a.html.

In this work, we study adversarial patrolling when a graph G_1 suddenly changes to G_2 due to edge deletions, changes in traversal times, or shifts in target values. The Defender is already executing a finite-memory randomized strategy σ_1 when the change occurs and must promptly replace it by a strategy σ_2 suited for G_2. The key observation is that recomputing a good strategy for G_2 is not sufficient: even if σ_1 is strong in G_1 and σ_2 is strong in G_2, the transition itself may create a short time window in which some target is effectively uncovered.

Specifically, if the change happens at time t, the Defender continues executing σ_1 until the first state reached at or after t, and only then transfers control to σ_2. The transfer is memory-aware: if σ_2 has a value-optimal memory state at the current vertex, the switch is immediate; otherwise, the controller may first move to a vertex from which σ_2 can start at full value. The attacker model is sharpened accordingly: in a changing environment, it may be optimal to attack in the middle of an edge traversal rather than only at departure times, see Fig. 9.

To quantify the vulnerability introduced by the switch, we define a *security hole* and propose an efficient algorithm to compute its upper bound. We show that only attacks launched within the last $d(\tau)$ time units before the switch can create a new vulnerability, so the analysis reduces to finitely many *steal* values parameterized by the attacked target, the current edge, and the offset time. These steals are then computed in batches by a heap-guided graph search with cached post-switch catch probabilities.

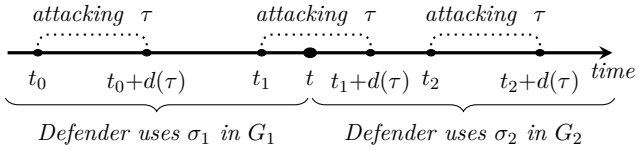


Figure 9: The coverage of an attack initiated at time t_1 short before the strategy switch can be very low due to the "incompatibility" of strategies σ_1 and σ_2.

The synthesis procedure is equally important. Instead of computing σ_2 from scratch, the method first adapts σ_1 to G_2 and then improves this warm start by local gradient optimization. To make this practical, we substantially re-engineered Regstar's optimization core. Forward-mode differentiation is replaced by reverse mode, the optimization loop is implemented in PyTorch with the Adam optimizer, and decaying noise is added to allow escaping weak local optima.

Empirically, on the office-building benchmarks previously used for Regstar, the new optimization procedure produces values tightly concentrated near the best values found, whereas the original outcomes are far more dispersed. Reverse-mode differentiation also reduces the backward-pass time by roughly three orders of magnitude.

In changing-environment experiments, the environment is perturbed by changing target costs, edge lengths, or removing edges. Across all small-change regimes, initializing the optimization in σ_1 yields higher Defender values and much smaller security holes than starting from a random strategy. For small changes, 50 optimization steps from σ_1 are typically not outperformed even by 400 steps from random initialization. Only for very large perturbations do random starts sometimes reach slightly higher values, but then the security holes are dramatically larger.

Overall, the paper makes a decisive step from strategy synthesis in a fixed graph to robust strategy adaptation under non-stationary adversarial conditions. Its most interesting ideas are the formalization of security holes, the reduction of their estimation to finitely many steal computations, and the use of warm-started differentiable optimization to adapt finite-memory patrolling strategies on realistic time scales.

My contributions. I reimplemented Regstar in Python using the existing core `C++` evaluation components, alongther with a PyTorch implementation of the optimization loop. I performed the experiments and analyzed the results. I contributed to the paper's conceptualization, presentation, and writing. (16%)

3.1.3 Memory Allocation for Finite-Memory Strategies

In this work, we address a practical bottleneck left open by earlier finite-memory synthesis methods. The paper was accepted at ICAPS 2026.

[9] Vojtěch Kůr, Vít Musil, and Vojtěch Řehák. "Memory Assignment for Finite-Memory Strategies in Adversarial Patrolling Games". In: *International Conference on Automated Planning and Scheduling*. 2026.

Recall that in regular patrolling strategies, the Defender operates on the state space of pairs (v, m), where v is a location and m is a memory value. Previous algorithms optimize strategies in this space, but they typically assume a fixed finite memory set M shared uniformly across all vertices. However, this is restrictive: uniform assignments quickly blow up the state space.

In contrast, we may allow different vertices to have different memory sizes, and the question is how to assign them across the graph. Clearly, handcrafted assignments require domain expertise and may fail when the environment changes. This work turns this design choice itself into an optimization problem.

We propose a general framework that separates *how* strategies are optimized from *how much memory* is allocated to each location. The method is deliberately black-box with respect to the patrolling model and the strategy optimizer. It only assumes that the strategy value decomposes over bottom strongly connected components and that, for every relevant attack, the attack value $D(e, \tau \mid \sigma)$ and its gradient can be computed. Under these assumptions, the routine can be plugged into existing synthesis methods. Hence, it is not a new solver for one objective, but a meta-method that upgrades an entire class of differentiable finite-memory tools.

The core algorithm starts from the uniform assignment $\mathrm{mem}_1(v) = 1$, runs any black-box optimizer to obtain a strategy σ_1, and then invokes an AdjustMemory procedure that proposes a refined assignment mem_2. The process iterates until the strategy value improves. The key idea is to inspect the near-maximal attacks and ask whether they "pull" a given state in incompatible gradient directions. If they do, then one memory state is insufficient, because a single local policy cannot react differently to these conflicting attack scenarios. The method also admits a practical variant. If the total number of states must remain below a budget L, the algorithm ranks attack profiles and retains only the most valuable ones.

In experiments, we integrate the routine with the best-performing available optimizers for variants of the patrolling games. The comparison is against uniform memory assignments and the expert-crafted *degree* assignment $\mathrm{mem}(v) = \mathrm{outdeg}(v)$, which is optimal for Eulerian-cycle-type patrols.

Experiments across four benchmark families confirm that uniform memory is rarely a good abstraction: it either cannot represent the optimum (too small) or makes the search space so large that good strategies are not found within the time limit. The automatic assignment consistently matches or outperforms both uniform and degree-based heuristics—reaching the optimum on Stars where degree fails, keeping memory small on Terrains where extra memory is harmful, and remaining competitive on Airports across heterogeneous instances.

Overall, the paper's contribution is to make finite-memory patrolling usable without expert-crafted memory assignments. It yields a simple, model-agnostic, and empirically robust outer loop around existing strategy optimizers, converting finite-memory synthesis from a partly manual art into a largely automated procedure.

My contributions. I co-supervised the project. I contributed to the paper's conceptualization, optimization, and evaluation design, and to its presentation and writing. (33%)

3.2 Long-Run Objectives in Stochastic Environments

3.2.1 Stability of Long-Run Objectives

Beyond patrolling games, we developed an optimization framework for long-run objectives in stochastic environments, which was accepted at IJCAI 2022.

[10] David Klaška, Antonín Kučera, Vít Musil, and Vojtěch Řehák. "General Optimization Framework for Recurrent Reachability Objectives". In: *Thirty-First International Joint Conference on Artificial Intelligence*. Vol. 5. July 2022, pp. 4642–4648. DOI: 10.24963/ijcai.2022/644.

In this work, we propose a general optimization language for infinite-horizon motion planning on graphs and Markov decision processes. The motivating problem is that classical long-run objectives based only on visit frequencies are often too weak: they can express mean payoff or average idleness, but they cannot control how irregularly visits are distributed in time. In surveillance, maintenance, or persistent monitoring, this matters because two strategies with the same frequency profile may differ dramatically in the variability of return times.

The key modeling idea is to define *recurrent reachability objectives* as closed-form expressions built from four atomic quantities: the expected hitting time from location v to a subset of configurations C, denoted $T(v \to C)$, its second moment $T^2(v \to C)$, the long-run frequency $F(e)$ of an edge, and the transition probability $p(e)$ assigned by the strategy. By combining these atoms with arithmetic operations, minima, maxima, and differentiable functions, one obtains a specification language that can express mean-payoff objectives, renewal-time objectives, adversarial and non-adversarial patrolling criteria, and variants that explicitly penalize variances or standard deviations. The need for such expressive power is immediate. We have already encountered the benefits of randomized strategies. However, if we control their quality only in expectation, we may still allow unintended behavior with very large local deviations.

On the complexity side, we show that deciding whether the optimum is below a given threshold is NP-hard. The framework therefore focuses on heuristic synthesis of high-quality finite-memory randomized strategies. The induced process is a finite Markov chain whose bottom strongly connected components determine the ergodic regimes relevant for infinite-horizon behavior. Objective values are evaluated on these components, and the global strategy value is obtained by taking the best one.

The algorithmic core is a gradient-based synthesis procedure. For a fixed component, the atomic quantities are computed as unique solutions of linear equation systems: hitting times and second moments come from Bellman-style equations, while edge frequencies arise from invariant-distribution equations. The remaining expressions are evaluated compositionally. The technical obstacle is that valid strategies live on products of simplices, and that the set of active edges changes discontinuously when a probability becomes zero. We resolve this by parameterizing strategies via softmax coefficients, applying a cutoff operator to recover sparse executable strategies, and relaxing the discontinuous objective into a smooth surrogate. Min and max are replaced by denser differentiable proxies.

We benchmarked the optimization of a weighted sum of mean payoff and its deviation, as well as a weighted sum of expected renewal time and its deviation. The results are consistent with theoretical predictions and show that the optimizer captures the intended trade-off between performance and stability tracing out a Pareto-curve.

My contributions. I contributed to the implementation, performed the experiments, and analyzed the results. I contributed to the paper's conceptualization, presentation, and writing. (25%)

3.2.2 Periodic Service and Maintenance

We also contributed to the non-adversarial setting through the following IJCAI 2023 article.

[11] David Klaška, Antonín Kučera, Vít Musil, and Vojtěch Řehák. "Mean Payoff Optimization for Systems of Periodic Service and Maintenance". In: *Thirty-Second International Joint Conference on Artificial Intelligence*. Vol. 5. Aug. 2023, pp. 5386–5393. DOI: 10.24963/ijcai.2023/598.

This work studies an infinite-horizon routing problem in which a service agent repeatedly visits nodes of a directed graph. Each node v has a payoff function $P_v(t)$ determined by the time t elapsed since its previous visit: servicing too early may yield partial reward, servicing in a preferred interval yields maximal reward, and servicing too late incurs a penalty. Together with traversal times and compulsory nodes, this defines a service specification. The objective is to maximize the long-run average payoff per unit of time. This model is closer to preventive maintenance than standard finite-horizon vehicle-routing formulations, because the relevant notion of quality is asymptotic regularity of returns rather than one-shift completion.

We first clarify the problem's difficulty. We showed that constructing an ε-optimal schedule is PSPACE-hard for every $\varepsilon \geq 0$, and that an optimal deterministic periodic schedule may require an exponentially long cycle. Thus, even though an optimal periodic solution exists, explicit cycle synthesis is algorithmically intractable. This is the conceptual point at which the paper departs from classical periodic routing and justifies the randomized approach based on finite-memory (RFM) strategies.

Given a strategy, the induced Markov chain decomposes into bottom strongly connected components, and the value of a strategy is the maximum mean payoff among these components. A strategy, together with the component on which the optimum is attained, is called an *RFM schedule*, which is a compact stochastic description of long-run service behavior. It can achieve substantially higher value than deterministic schedules with the same memory budget.

The algorithmic core is a differentiable optimization method for RFM schedules. Waiting times are encoded by splitting prolongable edges and representing waiting as repeated self-loops on auxiliary vertices, so that the controller can be parameterized continuously via softmax distributions over outgoing moves. For each bottom strongly connected component, the evaluation has two ingredients: the invariant distribution over augmented states, obtained from a linear system, and the expected payoff associated with first-return times to serviced nodes. The latter is computed by an evaluation procedure adapted from our earlier adversarial patrolling work.

Our second algorithmic contribution is a determinization procedure. We sample long routes from a synthesized RFM schedule and search them for high-quality repeated segments, which we interpret as candidate periodic cycles. The key insight is that a good RFM schedule acts as a compact generative model of promising long-run behavior: although it may use very little memory, sufficiently long samples contain cycles whose mean payoff exceeds the RFM value.

Experimental results show stable and practically relevant behavior. More importantly, the sampled periodic schedules have considerably better values than the optimized RFM values. Interestingly, very good periodic schedules can be sampled from intermediate RFM schedules rather than from converged ones, suggesting that the optimization process quickly discovers a promising region of the strategy space. Further optimization then soon reaches the limits of the RFM representation. This suggests that the RFM representation is a powerful abstraction for discovering long-horizon regularity by computationally cheap sampling.

Overall, the paper presents a convincing optimization pipeline for infinite-horizon service planning: first, synthesize a compact randomized finite-memory controller via differentiable programming, then determinize it via sampling high-quality cycles. The interesting point is not merely that randomization helps, but that it serves as an efficient intermediate representation for discovering long periodic schedules that are otherwise combinatorially out of reach.

My contributions. I contributed to the implementation, performed the experiments, and analyzed the results. I contributed to the paper's conceptualization, presentation, and writing. (25%)

4 List of Enclosed Publications

Here, we list the articles submitted as part of this habilitation thesis. They do not represent the applicant's entire research output.

[1] Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. "Differentiation of Blackbox Combinatorial Solvers". In: *The Eighth International Conference on Learning Representations*. ICLR 2020. May 2020. URL: https://openreview.net/forum?id=BkevoJSYPB.

[2] Anselm Paulus, Michal Rolínek, Vít Musil, Brandon Amos, and Georg Martius. "CombOptNet: Fit the Right NP-Hard Problem by Learning Integer Programming Constraints". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 8443–8453. URL: https://proceedings.mlr.press/v139/paulus21a.html.

[3] A. Paulus, G. Martius, and V. Musil. "LPGD: A General Framework for Backpropagation through Embedded Optimization Layers". In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. Proceedings of Machine Learning Research. PMLR, July 2024, pp. 39989–40014. URL: https://proceedings.mlr.press/v235/paulus24a.html.

[4] Michal Rolínek, Vít Musil, Anselm Paulus, Marin Vlastelica, Claudio Michaelis, and Georg Martius. "Optimizing Rank-based Metrics with Blackbox Differentiation". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Best paper nomination. Piscataway, NJ: IEEE, June 2020, pp. 7617–7627. DOI: 10.1109/CVPR42600.2020.00764.

[5] Michal Rolínek, Paul Swoboda, Dominik Zietlow, Anselm Paulus, Vít Musil, and Georg Martius. "Deep Graph Matching via Blackbox Differentiation of Combinatorial Solvers". In: *Computer Vision – ECCV 2020*. Vol. 28. Lecture Notes in Computer Science, 12373. Cham: Springer, Aug. 2020, pp. 407–424. DOI: 10.1007/978-3-030-58604-1_25.

[6] Anselm Paulus, Andreas René Geist, Pierre Schumacher, Vít Musil, Simon Rappenecker, and Georg Martius. "Differentiable Simulation of Hard Contacts with Soft Gradients for Learning and Control". In: *The Fourteenth International Conference on Learning Representations*. ICLR 2026. 2026. URL: https://openreview.net/forum?id=2EGtfFwxx8.

[7] David Klaška, Antonín Kučera, Vít Musil, and Vojtěch Řehák. "Regstar: Efficient Strategy Synthesis for Adversarial Patrolling Games". In: *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*. PMLR, Dec. 2021, pp. 471–481. URL: https://proceedings.mlr.press/v161/klaska21a.html.

[8] Tomáš Brázdil, David Klaška, Antonín Kučera, Vít Musil, Petr Novotný, and Vojtěch Řehák. "On-the-Fly Adaptation of Patrolling Strategies in Changing Environments". In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. PMLR, Aug. 2022, pp. 244–254. URL: https://proceedings.mlr.press/v180/brazdil22a.html.

[9] Vojtěch Kůr, Vít Musil, and Vojtěch Řehák. "Memory Assignment for Finite-Memory Strategies in Adversarial Patrolling Games". In: *International Conference on Automated Planning and Scheduling*. 2026.

[10] David Klaška, Antonín Kučera, Vít Musil, and Vojtěch Řehák. "General Optimization Framework for Recurrent Reachability Objectives". In: *Thirty-First International Joint Conference on Artificial Intelligence*. Vol. 5. July 2022, pp. 4642–4648. DOI: 10.24963/ijcai.2022/644.

[11] David Klaška, Antonín Kučera, Vít Musil, and Vojtěch Řehák. "Mean Payoff Optimization for Systems of Periodic Service and Maintenance". In: *Thirty-Second International Joint Conference on Artificial Intelligence*. Vol. 5. Aug. 2023, pp. 5386–5393. DOI: 10.24963/ijcai.2023/598.

The part containing the enclosed papers is excluded from the public version to prevent copyright infringement.