ITMO

NEURAL INFORMATION PROCESSING SYSTEMS

Learning Augmented Exact Exponential Algorithms

Tatiana Belova, Yuriy Dementiev, Danil Sagunov

CS Space

NeurIPS 2026

Learning-augmented exact search

LLMs

Statistical predictions about promising solutions

Learning-augmented algorithms

Use predictions to guide search with formal guarantees

Our focus: NP-hard combinatorial optimization

Provably faster exact search from noisy predictions

Speedup guarantees depend on the prediction model.

Why the exponential base matters

Which vertices can we delete to break every cycle?

$\approx 169\times$

2^{100} vs. 1.9^{100}

Illustration at 100 elements

One noisy bit per element

Fixed feasible target

1 0 1 0 0 1 0 0

1 means include the element

The solver may return any feasible subset.

One noisy prediction

1 0 0 1 0 1 0 1

3 bits differ in this example

Each bit is correct with probability $\geq \frac{1}{2} + \varepsilon$.

Search near the prediction

Example with 1,000 elements

≥60%

chance each bit is correct

At most 400 errors in expectation

450

maximum bit flips

Check all subsets within that radius

Exhaustive-search base drops from 2 to $2 - \Omega(\varepsilon^2)$.

≈0.087% of all subsets

Every answer is verified

A smaller base for monotone local search

Standard MLS

Try partial sets drawn from the full universe.

Complete each with an exact extension algorithm.

$2 - 1/c$

Exponential base

Prediction-guided MLS

Favor predicted members when building partial sets.

Use the same exact extension algorithm.

$2 - 1/c - \Omega_c(\varepsilon^2)$

Improved exponential base

c is the base of the extension algorithm's running time in the number of additions.

When the speedup applies

The instance has a feasible subset.

Known accuracy lower bound	Pairwise independent correctness High-probability success
Unknown accuracy	Mutually independent correctness Improved expected time with fallback

Feedback Vertex Set, d-Hitting Set, Min-Ones d-SAT, …

ИТМО

The next test for learned solvers

Learned solvers already predict partial assignments and score candidate elements.

Can a cheap predictor save search time on hard instances?

CS Space

NEURAL INFORMATION PROCESSING SYSTEMS