

Foundation Models for Accessible Symbolic Planning: Across Model Acquisition, Plan Generation, and Human Guidance

Pulkit Verma





How do we combine human expertise with machine speed and scale, especially when lives are on the line?

The Research Gap

- Automated planners are powerful but require expert knowledge.
- Humans have intuition, but can't solve complex problems fast enough.
- LLMs are unreliable for critical decision-making.

What if we could have a system where domain experts could guide AI planning in natural language and trust that the results are valid?

Can we make LLMs reliable planners
AND
use them to make planning accessible?

The Real Bottleneck in using Classical Planning

- A correct formal model of the world already exists.
- The system can produce a valid plan from it.
- A human expert can read and steer that formalism.

The Real Bottleneck in using Classical Planning

- A correct formal model of the world already exists.
Building the model needs a modeling expert.
- The system can produce a valid plan from it.
Planning reliably is exactly where LLMs fail today.
- A human expert can read and steer that formalism.
Steering the planner needs someone who writes PDDL.

Most Assumptions: Model given as input, user expert in PDDL

Reliable

PDDL-Instruct: Sound Plans, Assume model given

+ verifier

Accessible to Steer

Collaborative Planning: Steering without explicit PDDL

+ verifier

Accessible to Build

Vision2HTN: Hierarchy Learned

+ verifier

Least Assumptions: Flat Model given, Hierarchy Learned from Video

PDDL-Instruct: Enhancing Symbolic Planning Capabilities in LLMs through Logical Chain-of-Thought Instruction Tuning



Pulkit Verma^{*}



Ngoc La^{*}



Anthony Favier^{*}



Swaroop Mishra[†]



Julie A. Shah^{*}



ICAPS 2025 Workshop on Planning in the Era of LLMs (LM4Plan @ ICAPS25)

LLMs are good at Reasoning

Article

DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning

<https://doi.org/10.1038/s41586-025-09422-z>
Received: 14 February 2025
Accepted: 17 July 2025
Published online: 17 September 2025
Open access
Check for updates

General reasoning represents a long-standing and formidable challenge in artificial intelligence (AI). Recent breakthroughs, exemplified by large language models (LLMs)^{1,2} and chain-of-thought (CoT) prompting³, have achieved considerable success on foundational reasoning tasks. However, this success is heavily contingent on extensive human-annotated demonstrations and the capabilities of models are still insufficient for more complex problems. Here we show that the reasoning abilities of LLMs can be incentivized through pure reinforcement learning (RL), obviating the need for human-labelled reasoning trajectories. The proposed RL framework facilitates the emergent development of advanced reasoning patterns, such as self-reflection, verification and dynamic strategy adaptation. Consequently, the trained model achieves superior performance on verifiable tasks such as mathematics, coding competitions and STEM fields, surpassing its counterparts trained through conventional supervised learning on human demonstrations. Moreover, the emergent reasoning patterns exhibited by these large-scale models can be systematically used to guide and enhance the reasoning capabilities of smaller models.

Reasoning capability, the cornerstone of human intelligence, enables complex cognitive tasks ranging from mathematical problem-solving to logical deduction and programming. Recent advances in AI have demonstrated that LLMs can exhibit emergent behaviours, including reasoning abilities, when scaled to a sufficient size^{1,2}. However, achieving such capabilities in pre-training typically demands substantial

whereas unrestricted RL training can better incentivize the development of new reasoning capabilities in LLMs. Through this process, in the next section, our model (referred to as DeepSeek-R1) naturally develops diverse and sophisticated reasoning behaviours to solve reasoning problems, the model exhibits a tendency towards longer responses, incorporating verification, reflection and

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei Xuezhi Wang Dale Schuurmans Maarten Bosma
Brian Ichter Fei Xia Ed H. Chi Quoc V. Le Denny Zhou
Google Research, Brain Team
{jasonwei, dennyzhou}@google.com

Abstract

We explore how generating a *chain of thought*—a series of intermediate reasoning steps—significantly improves the ability of large language models to perform complex reasoning. In particular, we show how such reasoning abilities emerge naturally in sufficiently large language models via a simple method called *chain-of-thought prompting*, where a few chain of thought demonstrations are provided as exemplars in prompting. Experiments on three large language models show that chain-of-thought prompting improves performance on a range of arithmetic, commonsense, and symbolic reasoning tasks. The empirical gains can be striking. For instance, prompting a PaLM 540B with just eight chain-of-thought exemplars achieves state-of-the-art accuracy on the GSM8K benchmark of math word problems, surpassing even finetuned GPT-3 with a verifier.

Standard Prompting

Chain-of-Thought Prompting

nature medicine



Article

Toward expert-level medical question answering with large language models

Received: 14 June 2024
Accepted: 14 November 2024
Published online: 8 January 2025
Check for updates

A list of authors and their affiliations appears at the end of the paper

Large language models (LLMs) have shown promise in medical question answering, with Med-PaLM being the first to exceed a ‘passing’ score in United States Medical Licensing Examination style questions. However, challenges remain in how to form medical question answering and handling

Successive Prompting for Decomposing Complex Questions

Dheeru Dua¹ Shivanshu Gupta¹ Sameer Singh^{1,2} Matt Gardner¹
¹University of California, Irvine, USA ²Allen Institute for Artificial Intelligence
¹Microsoft Semantic Machines
{ddua, shivag5, sameer}@uci.edu, mattgardner@microsoft.com

Abstract
Answering complex questions that require making latent decisions is a challenging task, especially when limited supervision is available. Recent works leverage the capabilities of large language models (LLMs) to perform complex question answering in a few-shot setting by

Who kicked the longest field goal in the first half?
Q: What are all the field goals in the first half?
A: 12-yard, 42-yard and 33-yard
Q: What is the largest value in: 15, 20, 45, 30, and 10, 50, 25, 18, 35, 40, 12, 28, 32, 19, 41, 38, 22, 31, 27, 43, 36, 29, 44, 39, 24, 34, 26, 46, 40, 37, 47, 42, 33, 48, 41, 30, 51, 45, 32, 52, 49, 36, 53, 50, 43, 54, 51, 44, 55, 52, 47, 56, 54, 49, 57, 55, 50, 58, 56, 51, 59, 57, 52, 60, 59, 54, 61, 60, 55, 62, 61, 56, 63, 62, 57, 64, 63, 58, 65, 64, 59, 66, 65, 60, 67, 66, 61, 68, 67, 62, 69, 68, 63, 70, 69, 64, 71, 70, 65, 72, 71, 66, 73, 72, 67, 74, 73, 68, 75, 74, 69, 76, 75, 70, 77, 76, 71, 78, 77, 72, 79, 78, 73, 80, 79, 74, 81, 80, 75, 82, 81, 76, 83, 82, 77, 84, 83, 78, 85, 84, 79, 86, 85, 80, 87, 86, 81, 88, 87, 82, 89, 88, 83, 90, 89, 84, 91, 90, 85, 92, 91, 86, 93, 92, 87, 94, 93, 88, 95, 94, 89, 96, 95, 90, 97, 96, 91, 98, 97, 92, 99, 98, 93, 100, 99, 94, 101, 100, 95, 102, 101, 96, 103, 102, 97, 104, 103, 98, 105, 104, 99, 106, 105, 100, 107, 106, 101, 108, 107, 102, 109, 108, 103, 110, 109, 104, 111, 110, 105, 112, 111, 106, 113, 112, 107, 114, 113, 108, 115, 114, 109, 116, 115, 110, 117, 116, 111, 118, 117, 112, 119, 118, 113, 120, 119, 114, 121, 120, 115, 122, 121, 116, 123, 122, 117, 124, 123, 118, 125, 124, 119, 126, 125, 120, 127, 126, 121, 128, 127, 122, 129, 128, 123, 130, 129, 124, 131, 130, 125, 132, 131, 126, 133, 132, 127, 134, 133, 128, 135, 134, 129, 136, 135, 130, 137, 136, 131, 138, 137, 132, 139, 138, 133, 140, 139, 134, 141, 140, 135, 142, 141, 136, 143, 142, 137, 144, 143, 138, 145, 144, 139, 146, 145, 140, 147, 146, 141, 148, 147, 142, 149, 148, 143, 150, 149, 144, 151, 150, 145, 152, 151, 146, 153, 152, 147, 154, 153, 148, 155, 154, 149, 156, 155, 150, 157, 156, 151, 158, 157, 152, 159, 158, 153, 160, 159, 154, 161, 160, 155, 162, 161, 156, 163, 162, 157, 164, 163, 158, 165, 164, 159, 166, 165, 160, 167, 166, 161, 168, 167, 162, 169, 168, 163, 170, 169, 164, 171, 170, 165, 172, 171, 166, 173, 172, 167, 174, 173, 168, 175, 174, 169, 176, 175, 170, 177, 176, 171, 178, 177, 172, 179, 178, 173, 180, 179, 174, 181, 180, 175, 182, 181, 176, 183, 182, 177, 184, 183, 178, 185, 184, 179, 186, 185, 180, 187, 186, 181, 188, 187, 182, 189, 188, 183, 190, 189, 184, 191, 190, 185, 192, 191, 186, 193, 192, 187, 194, 193, 188, 195, 194, 189, 196, 195, 190, 197, 196, 191, 198, 197, 192, 199, 198, 193, 200, 199, 194, 201, 200, 195, 202, 201, 196, 203, 202, 197, 204, 203, 198, 205, 204, 199, 206, 205, 200, 207, 206, 201, 208, 207, 202, 209, 208, 203, 210, 209, 204, 211, 210, 205, 212, 211, 206, 213, 212, 207, 214, 213, 208, 215, 214, 209, 216, 215, 210, 217, 216, 211, 218, 217, 212, 219, 218, 213, 220, 219, 214, 221, 220, 215, 222, 221, 216, 223, 222, 217, 224, 223, 218, 225, 224, 219, 226, 225, 220, 227, 226, 221, 228, 227, 222, 229, 228, 223, 230, 229, 224, 231, 230, 225, 232, 231, 226, 233, 232, 227, 234, 233, 228, 235, 234, 229, 236, 235, 230, 237, 236, 231, 238, 237, 232, 239, 238, 233, 240, 239, 234, 241, 240, 235, 242, 241, 236, 243, 242, 237, 244, 243, 238, 245, 244, 239, 246, 245, 240, 247, 246, 241, 248, 247, 242, 249, 248, 243, 250, 249, 244, 251, 250, 245, 252, 251, 246, 253, 252, 247, 254, 253, 248, 255, 254, 249, 256, 255, 250, 257, 256, 251, 258, 257, 252, 259, 258, 253, 260, 259, 254, 261, 260, 255, 262, 261, 256, 263, 262, 257, 264, 263, 258, 265, 264, 259, 266, 265, 260, 267, 266, 261, 268, 267, 262, 269, 268, 263, 270, 269, 264, 271, 270, 265, 272, 271, 266, 273, 272, 267, 274, 273, 268, 275, 274, 269, 276, 275, 270, 277, 276, 271, 278, 277, 272, 279, 278, 273, 280, 279, 274, 281, 280, 275, 282, 281, 276, 283, 282, 277, 284, 283, 278, 285, 284, 279, 286, 285, 280, 287, 286, 281, 288, 287, 282, 289, 288, 283, 290, 289, 284, 291, 290, 285, 292, 291, 286, 293, 292, 287, 294, 293, 288, 295, 294, 289, 296, 295, 290, 297, 296, 291, 298, 297, 292, 299, 298, 293, 300, 299, 294, 301, 300, 295, 302, 301, 296, 303, 302, 297, 304, 303, 298, 305, 304, 299, 306, 305, 300, 307, 306, 301, 308, 307, 302, 309, 308, 303, 310, 309, 304, 311, 310, 305, 312, 311, 306, 313, 312, 307, 314, 313, 308, 315, 314, 309, 316, 315, 310, 317, 316, 311, 318, 317, 312, 319, 318, 313, 320, 319, 314, 321, 320, 315, 322, 321, 316, 323, 322, 317, 324, 323, 318, 325, 324, 319, 326, 325, 320, 327, 326, 321, 328, 327, 322, 329, 328, 323, 330, 329, 324, 331, 330, 325, 332, 331, 326, 333, 332, 327, 334, 333, 328, 335, 334, 329, 336, 335, 330, 337, 336, 331, 338, 337, 332, 339, 338, 333, 340, 339, 334, 341, 340, 335, 342, 341, 336, 343, 342, 337, 344, 343, 338, 345, 344, 339, 346, 345, 340, 347, 346, 341, 348, 347, 342, 349, 348, 343, 350, 349, 344, 351, 350, 345, 352, 351, 346, 353, 352, 347, 354, 353, 348, 355, 354, 349, 356, 355, 350, 357, 356, 351, 358, 357, 352, 359, 358, 353, 360, 359, 354, 361, 360, 355, 362, 361, 356, 363, 362, 357, 364, 363, 358, 365, 364, 359, 366, 365, 360, 367, 366, 361, 368, 367, 362, 369, 368, 363, 370, 369, 364, 371, 370, 365, 372, 371, 366, 373, 372, 367, 374, 373, 368, 375, 374, 369, 376, 375, 370, 377, 376, 371, 378, 377, 372, 379, 378, 373, 380, 379, 374, 381, 380, 375, 382, 381, 376, 383, 382, 377, 384, 383, 378, 385, 384, 379, 386, 385, 380, 387, 386, 381, 388, 387, 382, 389, 388, 383, 390, 389, 384, 391, 390, 385, 392, 391, 386, 393, 392, 387, 394, 393, 388, 395, 394, 389, 396, 395, 390, 397, 396, 391, 398, 397, 392, 399, 398, 393, 400, 399, 394, 401, 400, 395, 402, 401, 396, 403, 402, 397, 404, 403, 398, 405, 404, 399, 406, 405, 400, 407, 406, 401, 408, 407, 402, 409, 408, 403, 410, 409, 404, 411, 410, 405, 412, 411, 406, 413, 412, 407, 414, 413, 408, 415, 414, 409, 416, 415, 410, 417, 416, 411, 418, 417, 412, 419, 418, 413, 420, 419, 414, 421, 420, 415, 422, 421, 416, 423, 422, 417, 424, 423, 418, 425, 424, 419, 426, 425, 420, 427, 426, 421, 428, 427, 422, 429, 428, 423, 430, 429, 424, 431, 430, 425, 432, 431, 426, 433, 432, 427, 434, 433, 428, 435, 434, 429, 436, 435, 430, 437, 436, 431, 438, 437, 432, 439, 438, 433, 440, 439, 434, 441, 440, 435, 442, 441, 436, 443, 442, 437, 444, 443, 438, 445, 444, 439, 446, 445, 440, 447, 446, 441, 448, 447, 442, 449, 448, 443, 450, 449, 444, 451, 450, 445, 452, 451, 446, 453, 452, 447, 454, 453, 448, 455, 454, 449, 456, 455, 450, 457, 456, 451, 458, 457, 452, 459, 458, 453, 460, 459, 454, 461, 460, 455, 462, 461, 456, 463, 462, 457, 464, 463, 458, 465, 464, 459, 466, 465, 460, 467, 466, 461, 468, 467, 462, 469, 468, 463, 470, 469, 464, 471, 470, 465, 472, 471, 466, 473, 472, 467, 474, 473, 468, 475, 474, 469, 476, 475, 470, 477, 476, 471, 478, 477, 472, 479, 478, 473, 480, 479, 474, 481, 480, 475, 482, 481, 476, 483, 482, 477, 484, 483, 478, 485, 484, 479, 486, 485, 480, 487, 486, 481, 488, 487, 482, 489, 488, 483, 490, 489, 484, 491, 490, 485, 492, 491, 486, 493, 492, 487, 494, 493, 488, 495, 494, 489, 496, 495, 490, 497, 496, 491, 498, 497, 492, 499, 498, 493, 500, 499, 494, 501, 500, 495, 502, 501, 496, 503, 502, 497, 504, 503, 498, 505, 504, 499, 506, 505, 500, 507, 506, 501, 508, 507, 502, 509, 508, 503, 510, 509, 504, 511, 510, 505, 512, 511, 506, 513, 512, 507, 514, 513, 508, 515, 514, 509, 516, 515, 510, 517, 516, 511, 518, 517, 512, 519, 518, 513, 520, 519, 514, 521, 520, 515, 522, 521, 516, 523, 522, 517, 524, 523, 518, 525, 524, 519, 526, 525, 520, 527, 526, 521, 528, 527, 522, 529, 528, 523, 530, 529, 524, 531, 530, 525, 532, 531, 526, 533, 532, 527, 534, 533, 528, 535, 534, 529, 536, 535, 530, 537, 536, 531, 538, 537, 532, 539, 538, 533, 540, 539, 534, 541, 540, 535, 542, 541, 536, 543, 542, 537, 544, 543, 538, 545, 544, 539, 546, 545, 540, 547, 546, 541, 548, 547, 542, 549, 548, 543, 550, 549, 544, 551, 550, 545, 552, 551, 546, 553, 552, 547, 554, 553, 548, 555, 554, 549, 556, 555, 550, 557, 556, 551, 558, 557, 552, 559, 558, 553, 560, 559, 554, 561, 560, 555, 562, 561, 556, 563, 562, 557, 564, 563, 558, 565, 564, 559, 566, 565, 560, 567, 566, 561, 568, 567, 562, 569, 568, 563, 570, 569, 564, 571, 570, 565, 572, 571, 566, 573, 572, 567, 574, 573, 568, 575, 574, 569, 576, 575, 570, 577, 576, 571, 578, 577, 572, 579, 578, 573, 580, 579, 574, 581, 580, 575, 582, 581, 576, 583, 582, 577, 584, 583, 578, 585, 584, 579, 586, 585, 580, 587, 586, 581, 588, 587, 582, 589, 588, 583, 590, 589, 584, 591, 590, 585, 592, 591, 586, 593, 592, 587, 594, 593, 588, 595, 594, 589, 596, 595, 590, 597, 596, 591, 598, 597, 592, 599, 598, 593, 600, 599, 594, 601, 600, 595, 602, 601, 596, 603, 602, 597, 604, 603, 598, 605, 604, 599, 606, 605, 600, 607, 606, 601, 608, 607, 602, 609, 608, 603, 610, 609, 604, 611, 610, 605, 612, 611, 606, 613, 612, 607, 614, 613, 608, 615, 614, 609, 616, 615, 610, 617, 616, 611, 618, 617, 612, 619, 618, 613, 620, 619, 614, 621, 620, 615, 622, 621, 616, 623, 622, 617, 624, 623, 618, 625, 624, 619, 626, 625, 620, 627, 626, 621, 628, 627, 622, 629, 628, 623, 630, 629, 624, 631, 630, 625, 632, 631, 626, 633, 632, 627, 634, 633, 628, 635, 634, 629, 636, 635, 630, 637, 636, 631, 638, 637, 632, 639, 638, 633, 640, 639, 634, 641, 640, 635, 642, 641, 636, 643, 642, 637, 644, 643, 638, 645, 644, 639, 646, 645, 640, 647, 646, 641, 648, 647, 642, 649, 648, 643, 650, 649, 644, 651, 650, 645, 652, 651, 646, 653, 652, 647, 654, 653, 648, 655, 654, 649, 656, 655, 650, 657, 656, 651, 658, 657, 652, 659, 658, 653, 660, 659, 654, 661, 660, 655, 662, 661, 656, 663, 662, 657, 664, 663, 658, 665, 664, 659, 666, 665, 660, 667, 666, 661, 668, 667, 662, 669, 668, 663, 670, 669, 664, 671, 670, 665, 672, 671, 666, 673, 672, 667, 674, 673, 668, 675, 674, 669, 676, 675, 670, 677, 676, 671, 678, 677, 672, 679, 678, 673, 680, 679, 674, 681, 680, 675, 682, 681, 676, 683, 682, 677, 684, 683, 678, 685, 684, 679, 686, 685, 680, 687, 686, 681, 688, 687, 682, 689, 688, 683, 690, 689, 684, 691, 690, 685, 692, 691, 686, 693, 692, 687, 694, 693, 688, 695, 694, 689, 696, 695, 690, 697, 696, 691, 698, 697, 692, 699, 698, 693, 700, 699, 694, 701, 700, 695, 702, 701, 696, 703, 702, 697, 704, 703, 698, 705, 704, 699, 706, 705, 700, 707, 706, 701, 708, 707, 702, 709, 708, 703, 710, 709, 704, 711, 710, 705, 712, 711, 706, 713, 712, 707, 714, 713, 708, 715, 714, 709, 716, 715, 710, 717, 716, 711, 718, 717, 712, 719, 718, 713, 720, 719, 714, 721, 720, 715, 722, 721, 716, 723, 722, 717, 724, 723, 718, 725, 724, 719, 726, 725, 720, 727, 726, 721, 728, 727, 722, 729, 728, 723, 730, 729, 724, 731, 730, 725, 732, 731, 726, 733, 732, 727, 734, 733, 728, 735, 734, 729, 736, 735, 730, 737, 736, 731, 738, 737, 732, 739, 738, 733, 740, 739, 734, 741, 740, 735, 742, 741, 736, 743, 742, 737, 744, 743, 738, 745, 744, 739, 746, 745, 740, 747, 746, 741, 748, 747, 742, 749, 748, 743, 750, 749, 744, 751, 750, 745, 752, 751, 746, 753, 752, 747, 754, 753, 748, 755, 754, 749, 756, 755, 750, 757, 756, 751, 758, 757, 752, 7

Can we leverage LLMs' reasoning capabilities for Planning?

- What works for reasoning in LLMs?
- How to leverage it for planning?

Planning Domain Definition Language (PDDL)

```
(:action pickup
```

```
  :parameters (?ob)
```

```
  :precondition (and  
    (handempty)  
    (ontable ?ob))
```

```
  :effect (and  
    (not (handempty))  
    (not (ontable ?ob))  
    (holding ?ob))
```

```
)
```

→ **Precondition:** This condition must be true for this action to execute

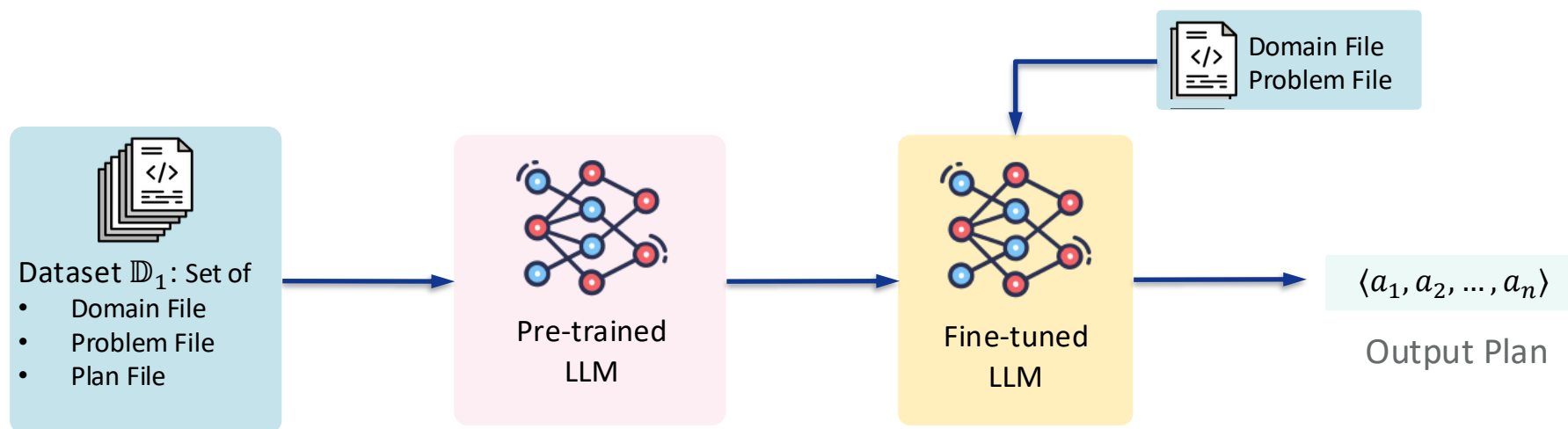
→ **Effect:** This is a set of conditions, one of which becomes true when this action is executed

What works for reasoning in LLMs?

- Finetuning
- Instruction tuning (finetuning with instructions)
- Chain-of-Thought prompting

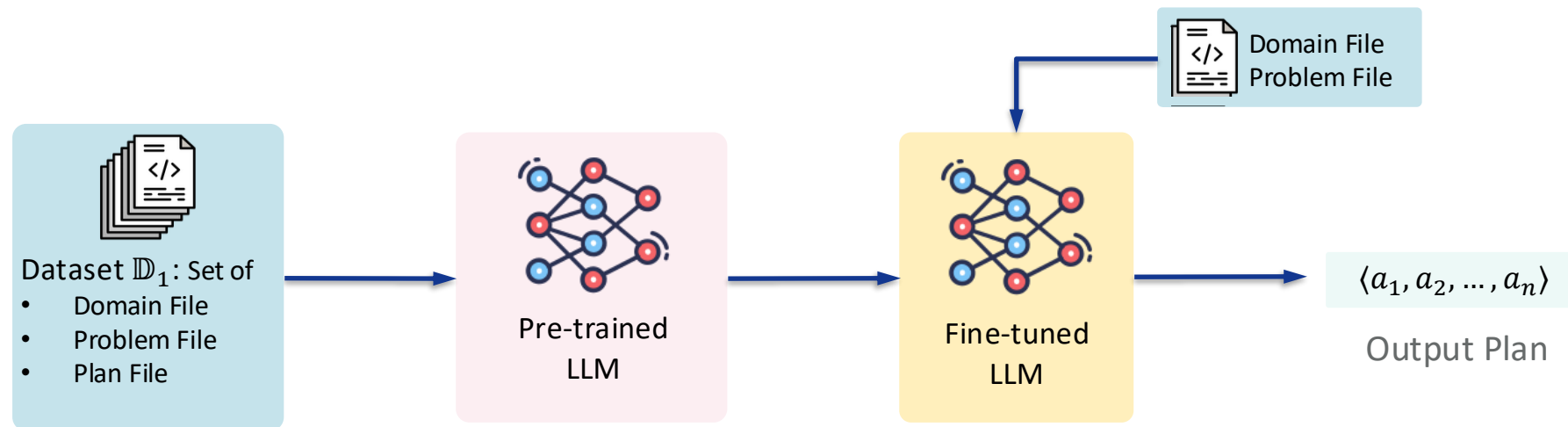
Finetuning

Adapt a pre-trained general LLM to excel at a specific task (planning) by training on domain examples.



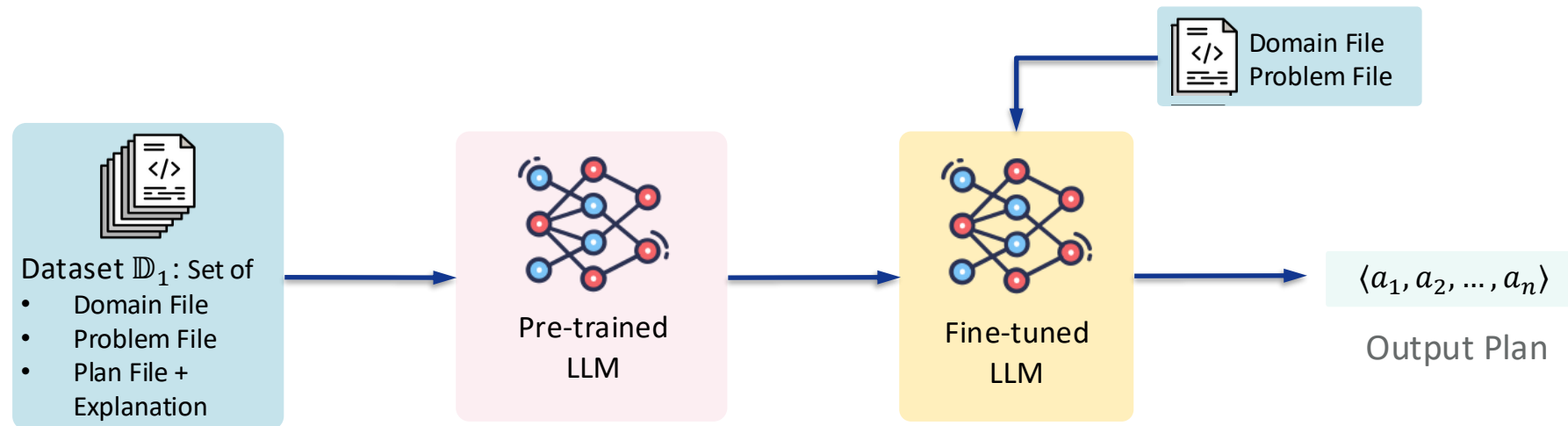
Finetuning with Negative Examples

Add some failing plans, label them as incorrect, and add them to the finetuning data.

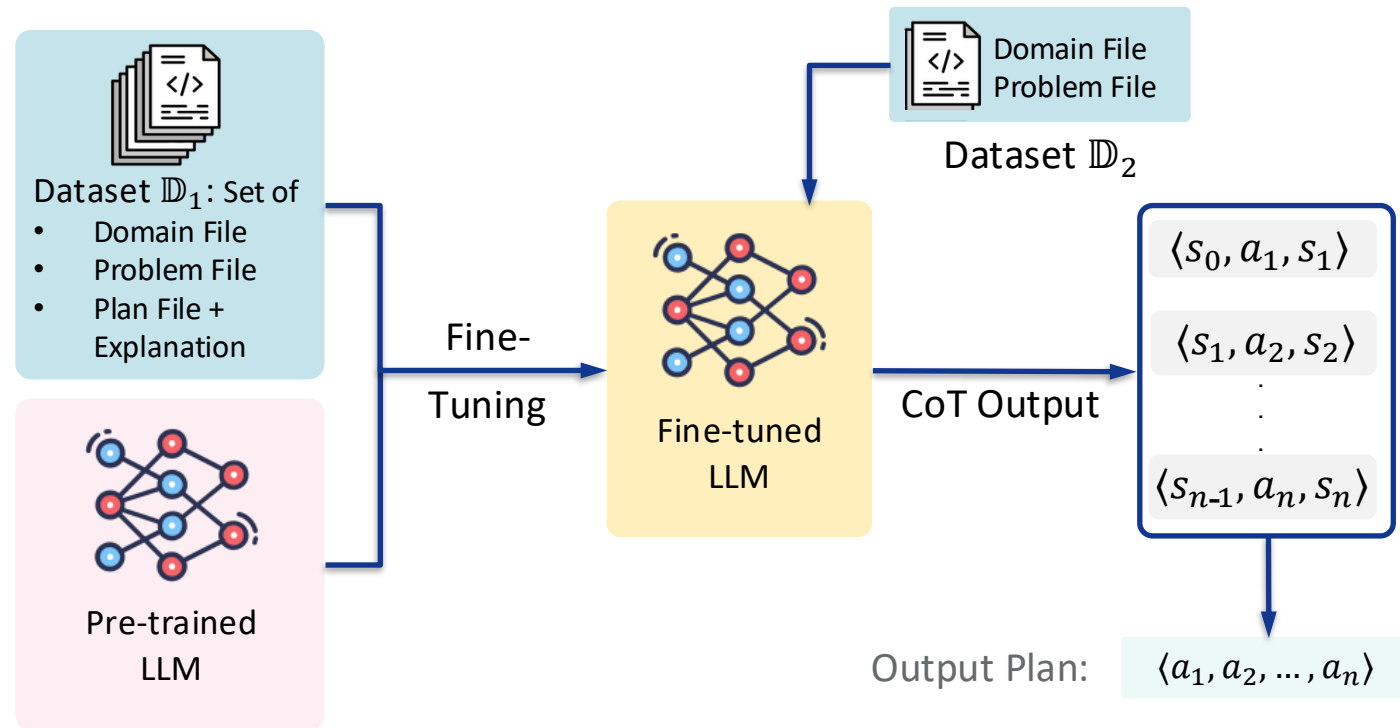


Instruction Finetuning

Add Explanations: Instructions teach the model WHAT planning means, not just PATTERNS in data. Tell it to check preconditions, apply effects, and verify goals.

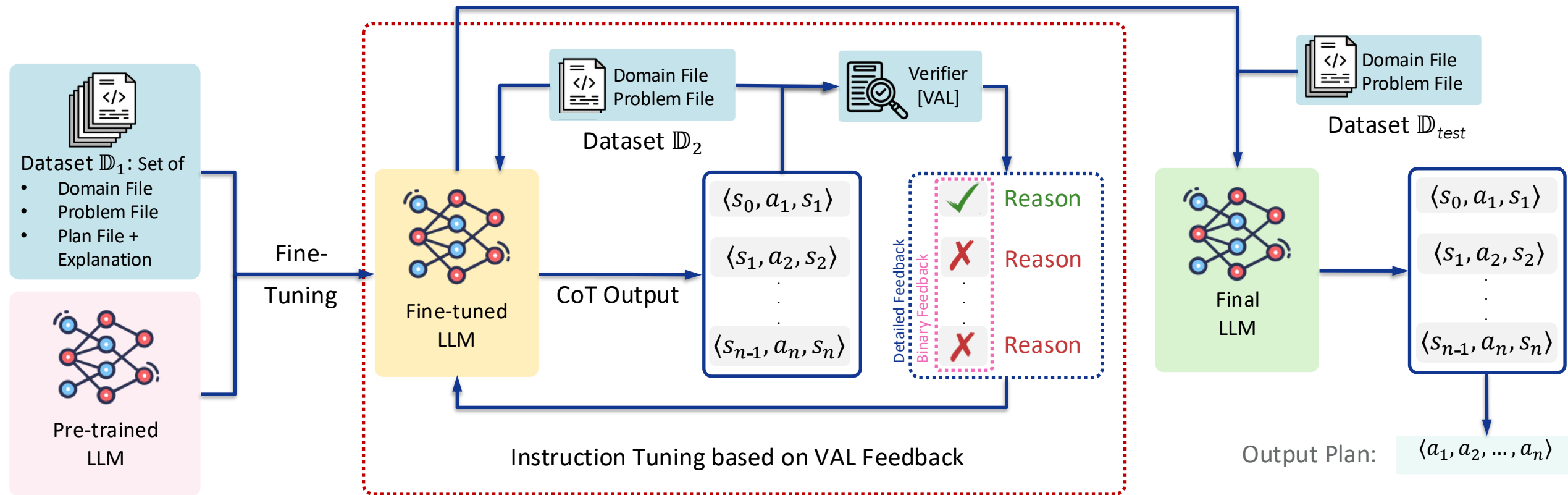


Augment finetuned LLM with Chain-of-Thought Prompting



Making the model show intermediate reasoning steps for planning instead of jumping to the final answer.

PDDLInstruct



Reasoning Chain Optimization

optimize the model parameters θ_t to
improve the generation of high-quality reasoning chains

$$\theta_t^r = \theta_t - \delta_1 \nabla_{\theta_t} L_{\text{reasoning}}(\theta_t, \mathbb{D}_{\text{reasoning}}^t)$$

$\{(s_{i-1}, a_i, s_i, f_i) : \forall \text{ steps in CoT plans generated at iteration } t\}$

loss function that measures the quality of the generated reasoning chains

Reasoning Chain Optimization

$$\mathcal{L}_{reasoning}(\theta_t, \mathbb{D}_{reasoning}^t) = \frac{1}{|\mathbb{D}_{reasoning}^t|} \sum_{(s_{i-1}, a_i, s_i, f_i) \in \mathbb{D}_{reasoning}^t} d(s_i, s_i^{expected}) + \lambda_{feedback} \mathcal{L}_{feedback}$$

$$\mathcal{L}_{feedback} = \begin{cases} 0 & \text{if action } a_i \text{ is valid} \\ \alpha_{pre} & \text{if precondition violation detected} \\ \alpha_{eff} & \text{if incorrect effect applied} \\ \alpha_{goal} & \text{if goal not achieved} \end{cases}$$

End-Task (Final) Performance Optimization

optimize from the reasoning-improved parameters θ_t^r to enhance overall planning

$$\theta_{t+1} = \theta_t^r - \delta_2 \nabla_{\theta_t^r} \mathcal{L}_{final}(\theta_t^r, \mathbb{D}_{final}^t)$$

$\{(d_j, p_j, \pi_i^t, v_j^t) : \forall \text{ problems } j \text{ at iteration } t\}$

loss function that measures how well the final outputs match the expected answers in the training data

Empirical Evaluation: Objectives

RQ1: Does logical CoT instruction tuning improve plan validity compared to standard approaches?

RQ2: How does the quality of feedback (binary vs. detailed) affect planning performance?

RQ3: How well does the approach generalize across different planning domains?

Empirical Evaluation: Dataset and Models

Three Domains:

- Blockworld
- Logistics
- Mystery Blocksworld

Three Models:

- Llama-3-8B
- GPT-4
- Gemma-3-270M

Benchmark

PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change

Karthik Valmeekam
School of Computing & AI
Arizona State University, Tempe.
kvalmeek@asu.edu

Matthew Marquez
School of Computing & AI
Arizona State University, Tempe.
mmarqu22@asu.edu

Alberto Olmo
School of Computing & AI
Arizona State University, Tempe.
aolmoher@asu.edu

Sarath Sreedharan*
Department of Computer Science,
Colorado State University, Fort Collins.
sarath.sreedharan@colostate.edu

Subbarao Kambhampati
School of Computing & AI
Arizona State University, Tempe.
rao@asu.edu

Logical CoT instruction tuning improves Plan Validity

Model	Domain	Baseline	Only P1	Only P2	PDDL-INSTRUCT				
				Detailed	Binary		Detailed		
				$\eta = 15$	$\eta = 10$	$\eta = 15$	$\eta = 10$	$\eta = 15$	
Llama-3	Blocksworld	28%	78%	72%	84%	89%	91%	94%	
	Mystery BW	1%	32%	17%	47%	49%	59%	64%	
	Logistics	11%	23%	45%	61%	72%	75%	79%	
GPT-4	Blocksworld	35%	41%	76%	79%	84%	87%	91%	
	Mystery BW	3%	17%	19%	39%	44%	54%	59%	
	Logistics	6%	27%	51%	64%	69%	72%	78%	
Gemma-3	Blocksworld	7%	12%	19%	37%	39%	54%	56%	
	Mystery BW	0%	2%	3%	22%	28%	24%	28%	
	Logistics	2%	13%	11%	18%	33%	27%	43%	

Detailed feedback is better than Binary Feedback

Model	Domain	Baseline	Only P1	Only P2	PDDL-INSTRUCT			
				Detailed	Binary		Detailed	
				$\eta = 15$	$\eta = 10$	$\eta = 15$	$\eta = 10$	$\eta = 15$
Llama-3	Blocksworld	28%	78%	72%	84%	89%	91%	94%
	Mystery BW	1%	32%	17%	47%	49%	59%	64%
	Logistics	11%	23%	45%	61%	72%	75%	79%
GPT-4	Blocksworld	35%	41%	76%	79%	84%	87%	91%
	Mystery BW	3%	17%	19%	39%	44%	54%	59%
	Logistics	6%	27%	51%	64%	69%	72%	78%
Gemma-3	Blocksworld	7%	12%	19%	37%	39%	54%	56%
	Mystery BW	0%	2%	3%	22%	28%	24%	28%
	Logistics	2%	13%	11%	18%	33%	27%	43%

PDDLInstruct's improved performance generalizes across domains

Model	Domain	Baseline	Only P1	Only P2	PDDL-INSTRUCT			
				Detailed	Binary		Detailed	
				$\eta = 15$	$\eta = 10$	$\eta = 15$	$\eta = 10$	$\eta = 15$
Llama-3	Blocksworld	28%	78%	72%	84%	89%	91%	94%
	Mystery BW	1%	32%	17%	47%	49%	59%	64%
	Logistics	11%	23%	45%	61%	72%	75%	79%
GPT-4	Blocksworld	35%	41%	76%	79%	84%	87%	91%
	Mystery BW	3%	17%	19%	39%	44%	54%	59%
	Logistics	6%	27%	51%	64%	69%	72%	78%
Gemma-3	Blocksworld	7%	12%	19%	37%	39%	54%	56%
	Mystery BW	0%	2%	3%	22%	28%	24%	28%
	Logistics	2%	13%	11%	18%	33%	27%	43%

Limitations

- Optimizing instruction tuning data.
- Fine-grained analysis of planning performance.
- Comparison with SoTA symbolic planners.
- Extending domain coverage.



A Collaborative Numeric Task Planning Framework based on Constraint Translations using LLMs



Anthony Favier



Ngoc La



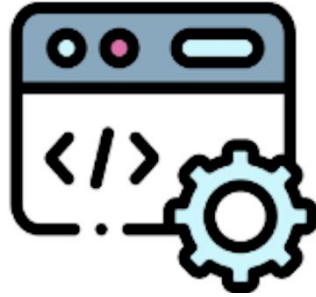
Pulkit Verma



Julie A. Shah



ICAPS 2025 Workshop on Planning in the Era of LLMs (LM4Plan @ ICAPS25)



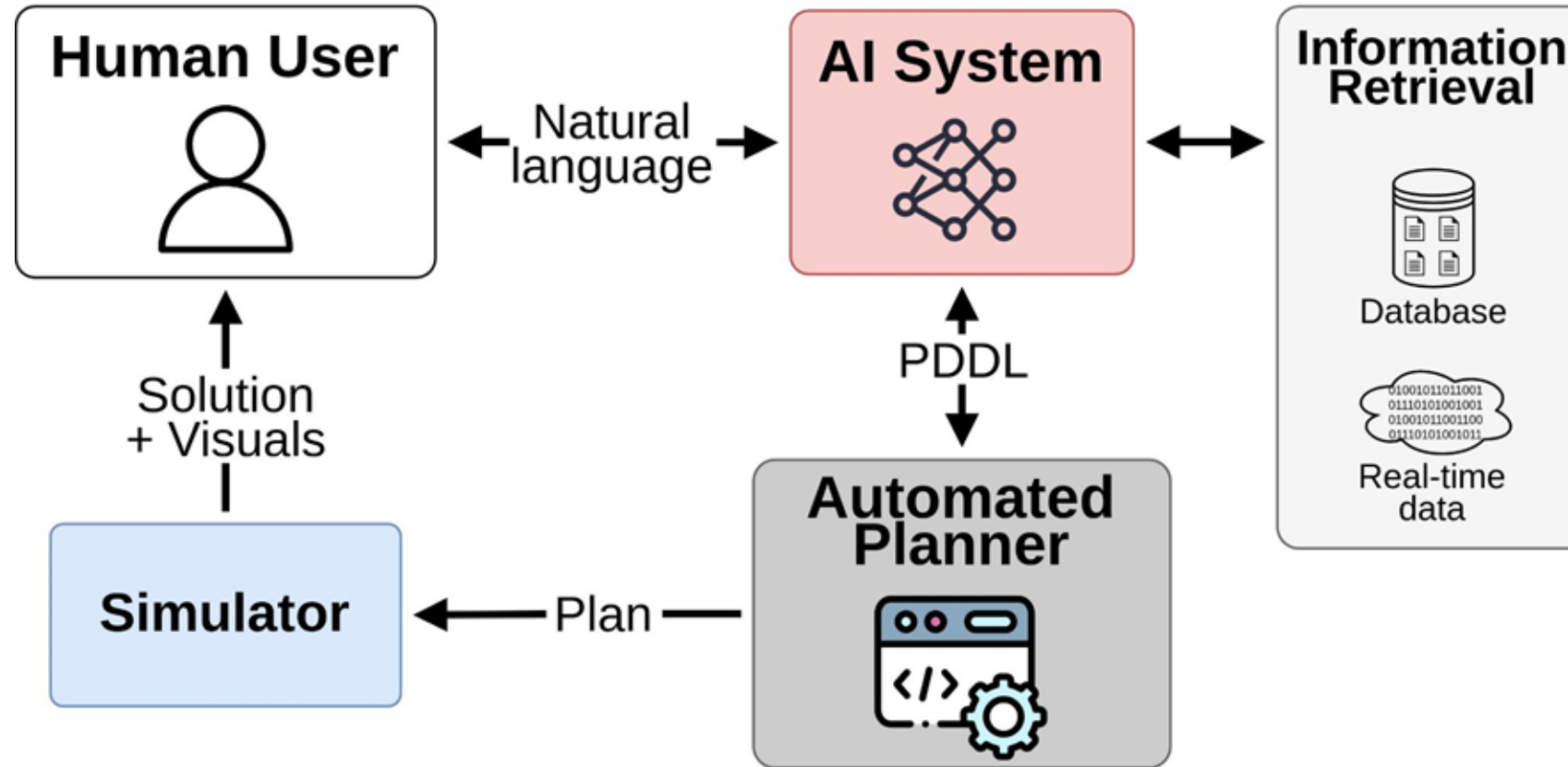
Formal Automated Planning

Requires:

- programming knowledge.
- technical expert interventions.



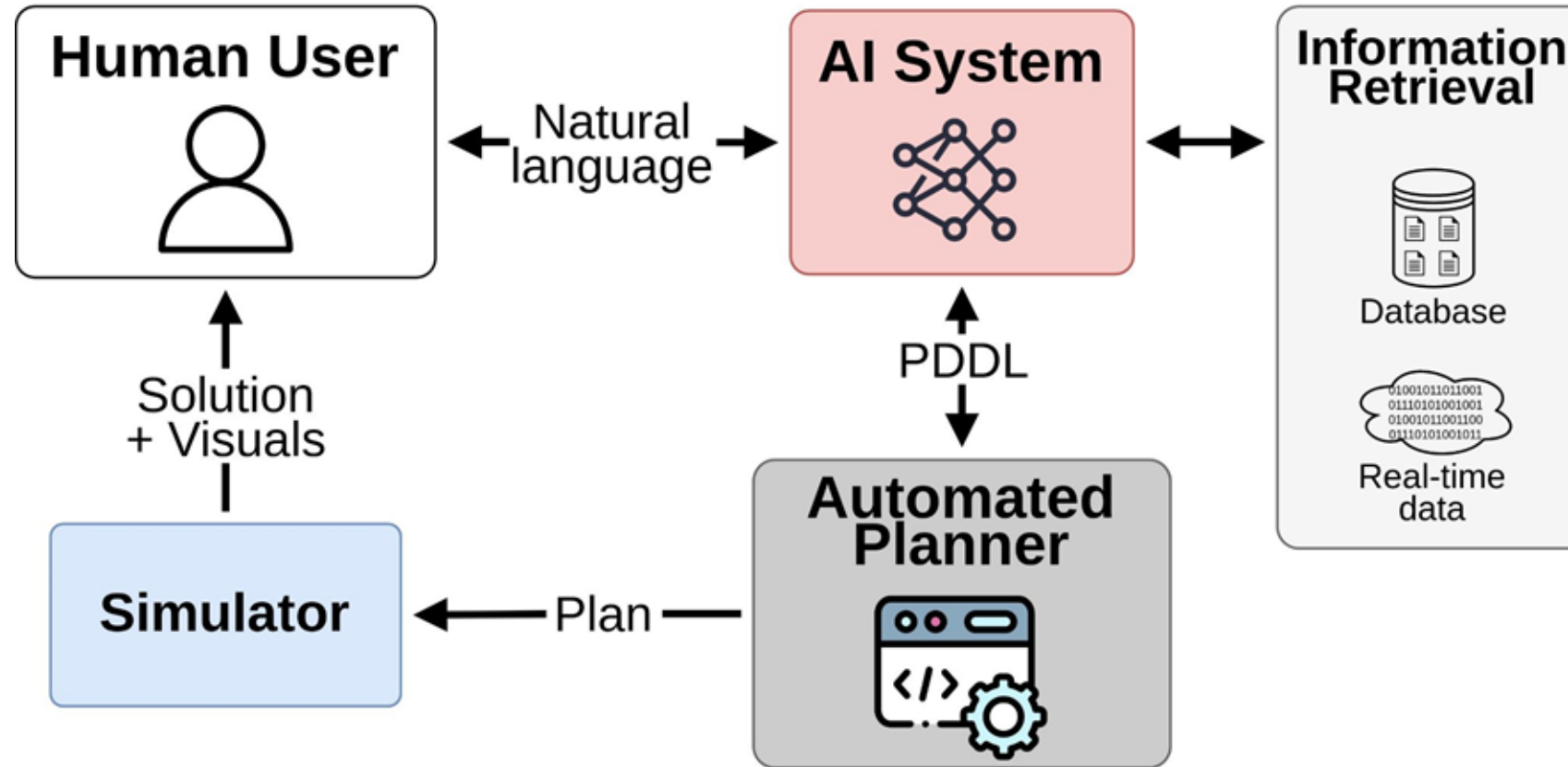
Hybrid Collaborative Planning Framework



A **symbolic** automated planner computes plans

An **LLM-based system** acts as an **interface** and for model elicitation

Hybrid Collaborative Planning Framework



Human can **influence** problem solving,
without technical expertise requirements

No constraints

Setting: DEFAULT
Planning mode: anytime, TO=15.0
Problem (zenoreal):

- NumericTCORE/benchmark/ZenoTravel-no-constraint/domain_with_n.pddl
- PDDL/zenoreal.pddl

=== ADDING CONSTRAINT ===

Enter your constraint:

1) Human input

Elapsed Time: 0.0 s

Confirm

Translate

Risk Analysis

Chat

Plan

Plans

Previous:

None

Current:

None

R0 - Only use plane1

- D1- Plane2 cannot board any passengers
- D2- Plane2 cannot debark any passengers
- D3- Plane2 cannot fly slow between any cities
- D4- Plane2 cannot fly fast between any cities
- D5- Plane2 cannot refuel
- D6- Plane3 cannot board any passengers
- D7- Plane3 cannot debark any passengers
- D8- Plane3 cannot fly slow between any cities
- D9- Plane3 cannot fly fast between any cities
- D10- Plane3 cannot refuel

2) Added Constraints

- Plane3 cannot fly slow between any cities
- Plane3 cannot fly fast between any cities
- Plane3 cannot refuel

Are you satisfied with the decomposition? If not, provide any desired feedback or type 'explain'.

User: yes

Encoding ...

Elapsed Time: 0.0 s

Confirm

Translate

Risk Analysis

Chat

Plan

Plans

Previous:

None

Current:

None

R0 - Only use plane1

- D1- Plane2 cannot board any passengers
- D2- Plane2 cannot debark any passengers
- D3- Plane2 cannot fly slow between any cities
- D4- Plane2 cannot fly fast between any cities
- D5- Plane2 cannot refuel
- D6- Plane3 cannot board any passengers
- D7- Plane3 cannot debark any passengers
- D8- Plane3 cannot fly slow between any cities
- D9- Plane3 cannot fly fast between any cities
- D10- Plane3 cannot refuel

3) Plan

- PDDL/zenoreal.pddl

Constraints loaded

=== PLANNING ===

Compiling ... OK [1.52s]

Planning (anytime, TO=15.0s) ... OK [15.06s]

Plans

Previous:

None

Current:

Plan-Length: 48
Metric: 15536.0
Planning time: 15.06
Found Plan:
0.0: (refuel_plane1)
1.0: (board_person4_plane1_boston)
2.0: (flyfast_plane1_boston_washington)
3.0: (board_person2_plane1_washington)
4.0: (board_person8_plane1_washington)
5.0: (flyslow_plane1_washington_boston)
6.0: (refuel_plane1)
7.0: (flyslow_plane2_washington_washington)
8.0: (flyslow_plane1_boston_dallas)
9.0: (board_person9_plane1_dallas)
10.0: (flyfast_plane1_dallas_seattle)
11.0: (debark_person9_plane1_seattle)
12.0: (refuel_plane1)
13.0: (flyslow_plane1_seattle_denver)
14.0: (debark_person4_plane1_denver)
15.0: (flyslow_plane1_denver_washington)
16.0: (refuel_plane1)
17.0: (flyslow_plane1_washington_seattle)
18.0: (flyslow_plane1_seattle_dallas)

Elapsed Time: 16.5 s

Confirm

Translate

Risk Analysis

Chat

Plan



ZenoR

PDSim



4) Simulation

Plan Actions

refuel (plane1)

flyslow (plane1, boston, denver)

flyfast (plane1, denver, washington)

refuel (plane1)

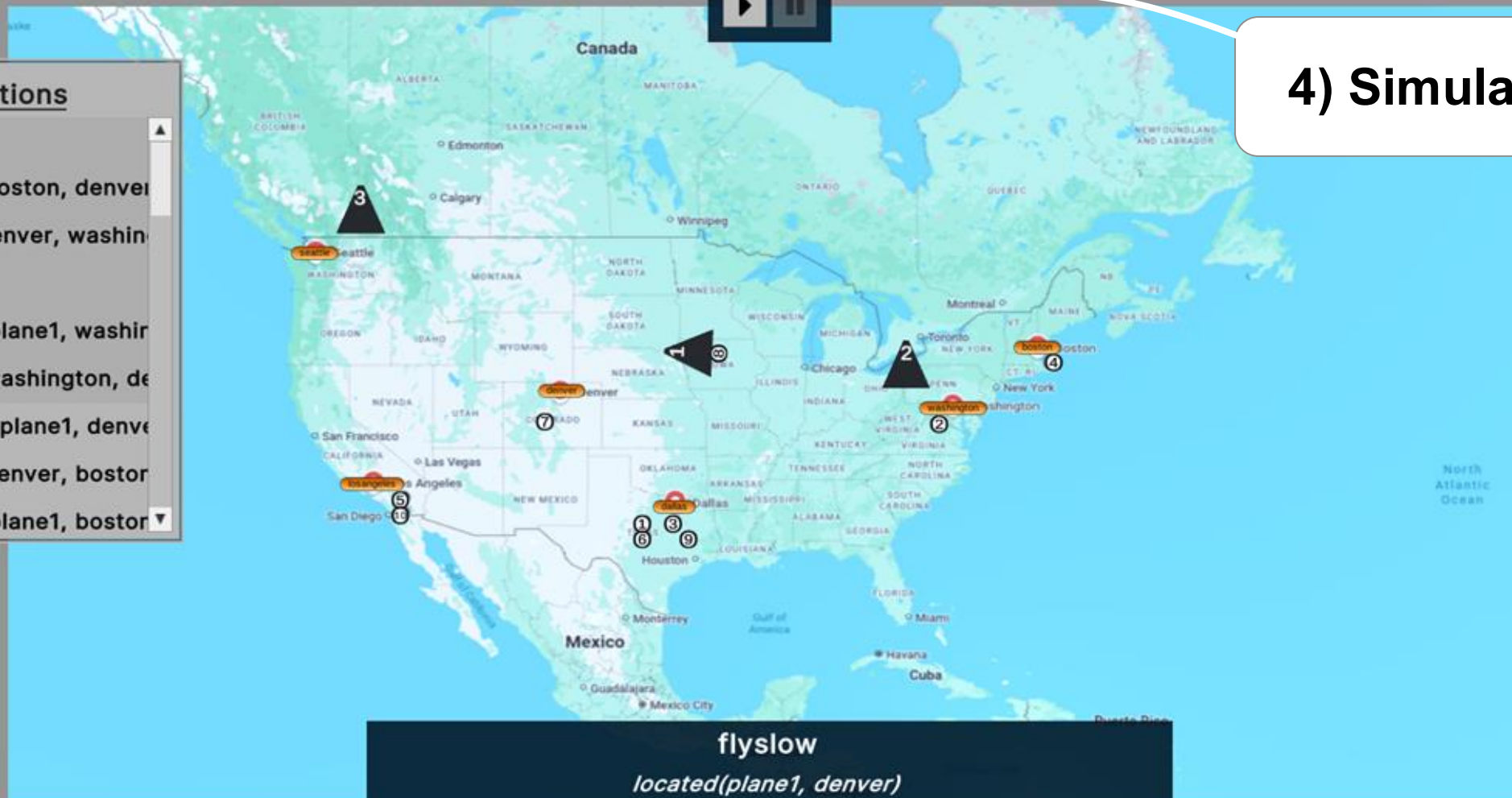
board (person8, plane1, washington)

flyslow (plane1, washington, denver)

debark (person8, plane1, denver)

flyslow (plane1, denver, boston)

board (person4, plane1, boston)



flyslow

located(plane1, denver)

Plan Panel

Action Tab

Speed Controls

Object Info Panel

Camera Controls

Elapsed Time: 16.5 s

Confirm

Translate

Risk Analysis

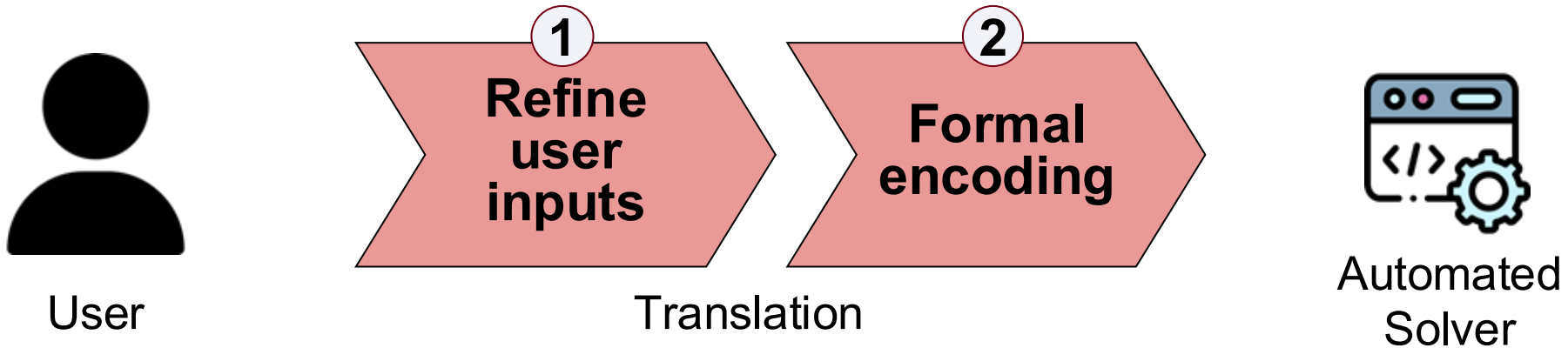
Chat

Plan

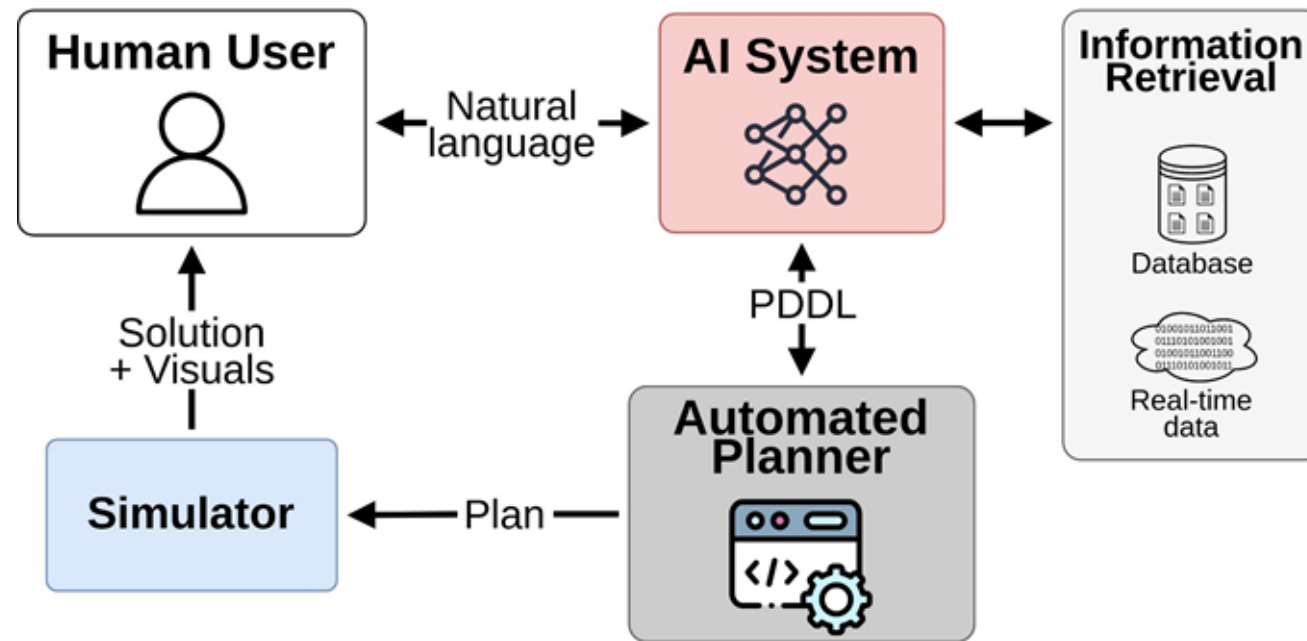
Guidance Translation

Translate user inputs as *guidance* for the *solver*

Two-step process:



Hybrid collaborative planning framework (neuro-symbolic)



Creates a **collaborative, mixed-initiative planning** scheme where the human can influence problem solving, **without** technical expertise requirements

Learning HTNs from Visual Demonstration with Vision-Language Models: Preliminary Results



Ngoc La



Karthik Mahadevan



Pulkit Verma

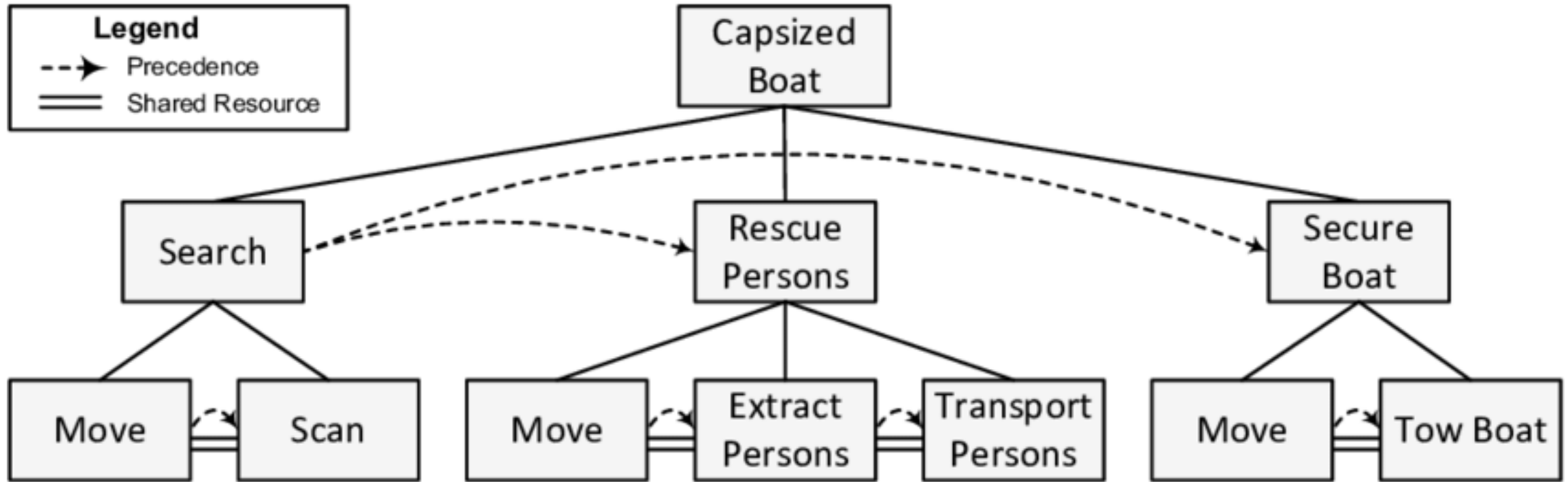


Julie A. Shah



ICAPS 2026 Workshop on Hierarchical Planning (HPlan)

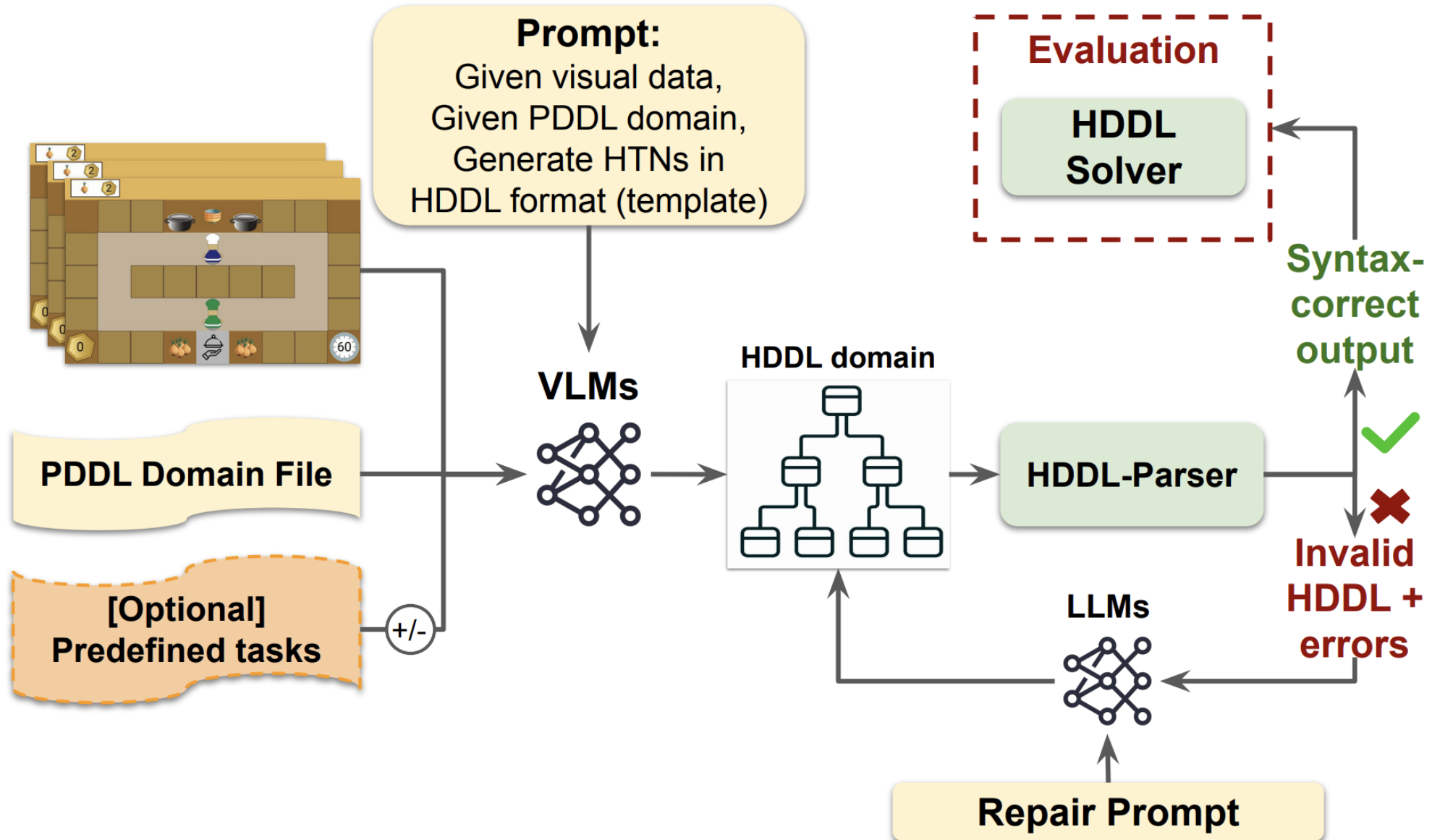
Hierarchical Task Networks (HTNs)



Where does the hierarchical model come from?

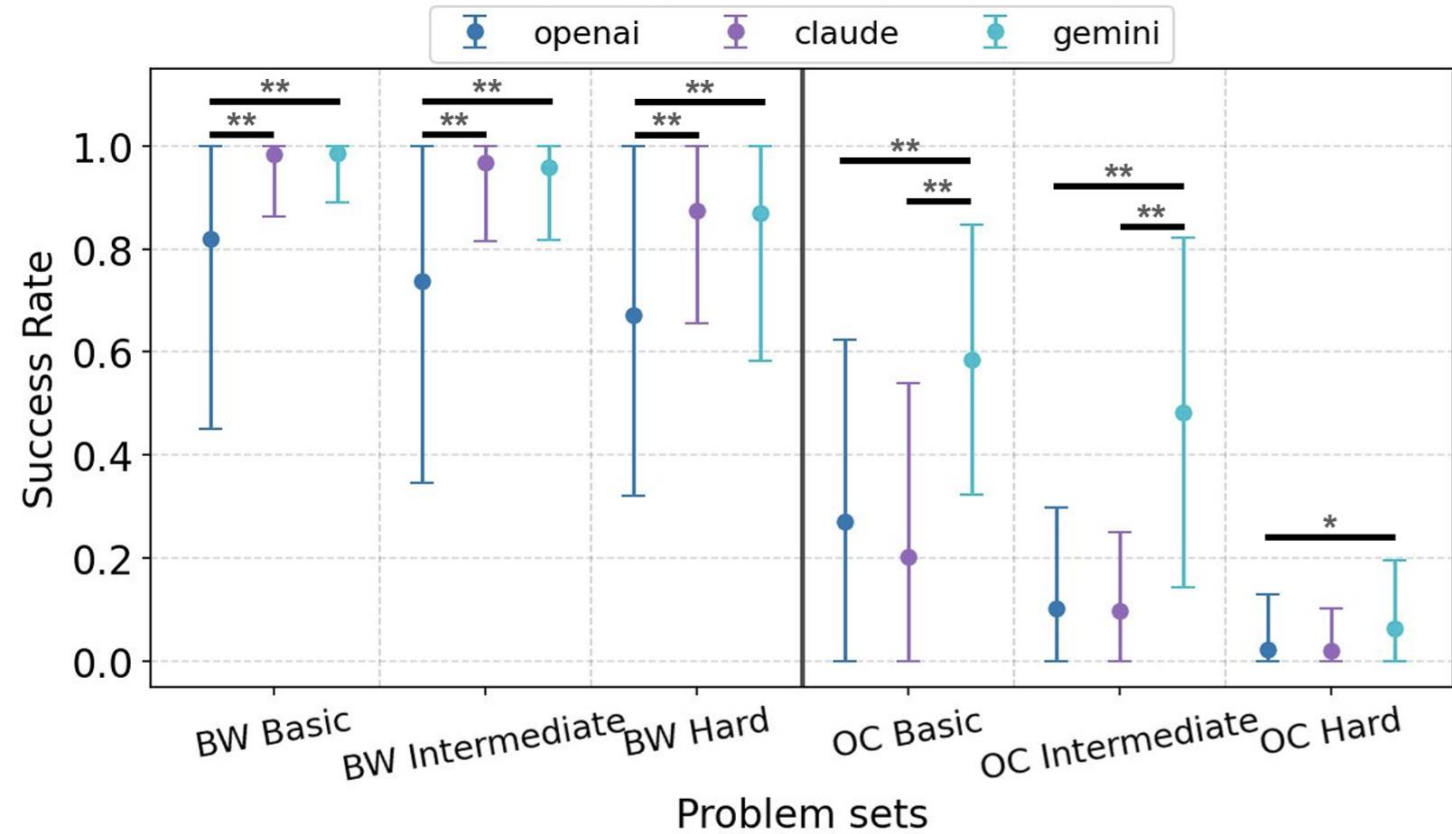
- HTNs encode human strategy as task hierarchies
 - powerful for long-horizon planning
- The flat actions can be modelled, but the hierarchy must be hand-authored
 - needs real expertise
- Prior HTN learning uses symbolic plan traces; learning the hierarchy from raw video is unexplored
- Our question: given a flat PDDL model, can we learn the strategy hierarchy from how people demonstrate?

Vision2HTN



Results

- Tested on Blocksworld and Overcooked (multi-agent); varied task priors, visual input, and VLM
- Task priors help most where structure is hard to infer (big lift in Overcooked, little in Blocksworld)
- Visual input is domain-dependent: Blocksworld did better without video (already in pretraining); Overcooked needed video to recover collaborative methods
- Model matters, non-uniformly: Claude: richer hierarchies, wins Blocksworld; Gemini: leaner, wins Overcooked



Foundation Models for Accessible Symbolic Planning: Across Model Acquisition, Plan Generation, and Human Guidance

Pulkit Verma

