ChinaTravel: An Open-Ended Travel Planning Benchmark with Compositional Constraint Validation for Language Agents

Jie-Jing Shao*, **Bo-Wen Zhang***, **Xiao-Wen Yang***, **Bai-Zhi Chen**, **Si-Yu Han**, **Jing-Hao Pang**, **Wen-Da Wei**, **Guohao Cai**, **Zhenhua Dong**, **Lan-Zhe Guo†**, **Yu-Feng Li†**

State Key Laboratory for Novel Software Technology, Nanjing University, China

Noah's Ark Lab, Huawei, China * Equal Contribution †Corresponding Authors

ICLR April 23rd - 27th, 2026

Project Page | Codebase | Dataset | WeChat

TL;DR

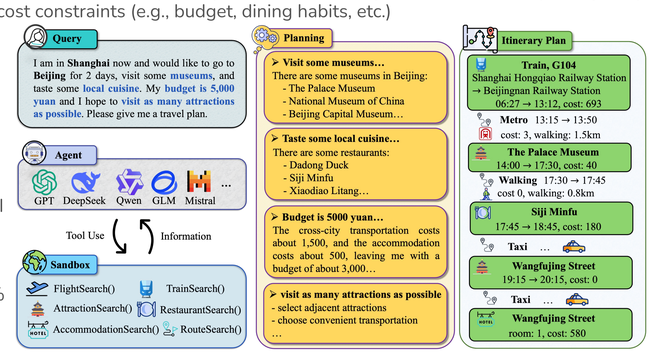
The first open-ended travel-planning benchmark grounded in **1,154 authentic Chinese human queries**, with a **compositional DSL that scales constraint validation beyond slot-filling**. Neuro-Symbolic agents are **10× more reliable** than pure LLMs, yet still far from robust compositional generalization.

Welcome to China • Bem-vindos à China • 中国欢迎您

Let language agents plan your trip.

Background | Travel Planning Agent

- **Task Specification:** Given an open-ended query, agents integrate information from multiple tools (flights, trains, hotels, restaurants, attractions, maps) to produce a multi-day, multi-POI itinerary that jointly satisfies spatial, temporal, and cost constraints (e.g., budget, dining habits, etc.)
- **Practical Value:**
  - A laborious daily task
  - 5+ apps:
  - Hours of cross-platform information search & booking
  - Strong demand for AI agents
- **Academic Challenge:**
  - Long-horizon, multi-day & POI
  - Spatial / temporal / cost interdependencies
  - Open-ended NL intents
  - GPT-4o / DeepSeek-V3: ~ 0%
- **Widespread Attention:**
  - Frontier AI Labs:
  - Verticals:



Research Gap | From Slot-Filling to Compositional Instructions

- **Prior Paradigm, Slot-Filling:**
  - Extract fixed attributes: budget, date, POI info …
  - Closed intent set over propositional predicates
  - Rule-based verification, no unseen-constraint coverage
- **Near Saturation, Misleading Progress:**
  - NeSy[1] 97% vs. LLM-only 4% on templated TravelPlanner[2]
  - Saturated within a closed, propositional constraint space
  - Hides true bottleneck: compositional generalization
- **Our Paradigm Shift, Compositional Instructions:**
  - Atomic concepts (cost / time / type / dist) × logic primitives
  - Arbitrary composable constraints and DSL-verifiable 100+ constraint types in the evaluation from authentic Chinese human queries, 85 unseen during the development.

Templated Synthetic Data in Previous Work

Our **budget is set at $1,400** for this trip and we require our **accommodations to be visitor-friendly**. We would like to have options to **dine at Indian, American, Chinese, and Italian restaurants**. We also prefer **not to self-drive**.

Our **budget is set at $3,100**. We require **accommodations that are pet-friendly** and we would prefer to **have entire rooms** to ourselves. We **do not plan on self-driving** for trip.

Authentic Data with Open-Ended Requirements

The budget of dining is ¥1000.
The budget excluding flights is ¥3000.
Arriving Beijing before 13PM.
Visit the Great Wall at the second day.

Compositional Constraints

[1] Large Language Models Can Solve Real-World Planning Rigorously with Formal Verification Tools. NAACL'25
[2] TravelPlanner: A Benchmark for Real-World Planning with Language Agents. ICML'24

Sandbox | Real Travel Information with Tool APIs

10	12K+	6.5K	6
Chinese Cities	POIs	Flights + Trains	Env. categories

- **Cities:** Beijing · Shanghai · Chengdu · Chongqing · Guangzhou · Hangzhou · Nanjing · Shenzhen · Suzhou · Wuhan.
- **Commercial-Style APIs:** attribute-select, keyword-search, nearest-neighbor, is-open-at-time, cuisine/recommended-food, intercity-transport, etc.
- **6 env. categories:** Dietary · Accommodation · Transportation · Attraction · Temporal · Spatial.

Compositional Constraint Validation | DSL

- **Domain-Specific Language, Atomic × Primitives:**
  - Atomic concepts: cost, time, location, cuisine type, hotel feature, distance …
  - Primitives: boolean, arithmetic, set, for-enumerate, if-effect …
  - Python-like Syntax
- **Executable and Scalable:**
  - Each requirement compiles to Python, validated by a standard compiler
  - Composes over unseen constraint types.
  - Zero per-requirement rule engineering

Name	Syntax	Description
variables	$x, y, z, \cdots$	Variables that refer to activities in the travel planning domain.
not	$not\ expr$	The negation of an Boolean-valued expression.
and,or	$expr_1\ and\ expr_2$	The conjunction/disjunction of an Boolean-valued expression.
<,>,==	$expr_1 < expr_2$	Return an expression with built-in number comparison functions.
+,−,*,/	$expr_1 + expr_2$	Return an expression with built-in number calculation functions.
attributes	$cost(var)$	A function that takes activities as inputs and returns the attributes, such as cost, type or time.
relation	$dist(expr_1, expr_2)$	A function that takes locations as inputs and returns the distance.
effect	$var = expr$	An assignment affects a variable var with the expression $expr$.
union, inter, diff	$uni(\{var\}_1, \{var\}_2)$	Return a set with the built-in union/intersection/difference operations of given two sets.
enumerate	*for var in* $\{var\}$	Enumerate all variables in the collection $\{var\}$.
when	*if* $expr$: *effect*	The conditional effect takes a Boolean-valued condition of the expression $expr$, and the effect *effect*.

ChinaTravel's Domain-Specific Language (DSL)

E.g.

```
# Dining expenses <= 1000 CNY.
dining_cost = 0
for act_i in allactivities(plan):
  typ = activity_type(act_i)
  if typ=="breakfast" or typ=="lunch"
or typ=="dinner": dining_cost =
dining_cost + activity_cost(act_i)
return dining_cost <= 1000
```

(a) Dining expenses.

```
# Arriving in Shanghai should be before
6 PM on the second day.
return_time = 0
for act_i in day_activities(plan, 2):
  typ = activity_type(act_i)
  dest = transport_destination(act_i)
  if (typ=="train" or typ=="airplane")
and des=="Shanghai": return_time ==
activity_endtime(act_i)
return return_time < "18:00"
```

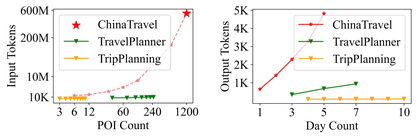
(b) Arrived Time.

```
# The number of attractions visited
count = 0
for act_i in all_activities(plan):
  if
activity_type(act_i)=="attraction":
count = count + 1
return count
# Compare the return during evaluation
of preference ranking
```

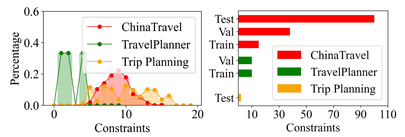
(c) Count of attraction visited.

Benchmark Characteristics | Challenges

- **Context-Rich Long-Horizon Planning:**
  - **540M contextual tokens/query** from dense POI graphs, exceeding DeepSeek-V3 (64K) and GPT-4o (128K); 6-POI down-sampling still leaves 40K. 1,200+ candidate POIs per query (4× TravelPlanner, 120× Trip Planning).
  - **Output 4.8K tokens for 5-day itineraries**, vs. TravelPlanner's 0.9K (7-day) and Trip Planning's 0.5K (30-day).

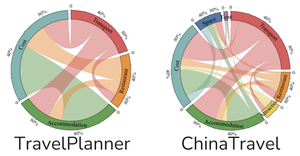


Input / Output Tokens Explosion

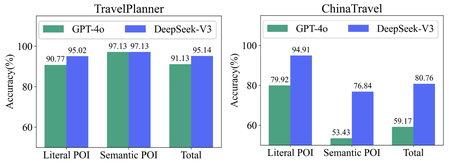


Open-ended Constraints

- **Diversity and Openness of Travel Requirements:**
  - Gaussian 6 to 12 constraints per query vs. TravelPlanner's ≤5.
  - Dev 15 → Eval 100 constraint types, 85 novel compositions unseen during development.
  - Zipf long-tail co-occurrence pattern, with cost × transport/hotel correlation matching common sense, contrasting TravelPlanner's uniform synthetic distribution.



Co-occurrence distribution of different constraints

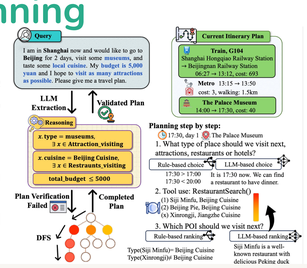


Intent Grounding

- **Contextual Grounding for Implicit Intent:**
  - Human queries often express intent implicitly, producing ambiguity beyond DB attributes. e.g., "local cuisine" grounds to Beijing cuisine in BJ but Benbang in SH; "traveling with children who cannot eat spicy" means excluding Sichuan and Chongqing cuisines.
  - We design an Intent Grounding auxiliary task: mask POIs in DSL constraints with ⟨placeholder⟩ tags and let LLMs fill them from context. Targets are labeled Literal (explicit) or Semantic (cultural / contextual).
  - On ChinaTravel, GPT-4o drops 79% → 53% and DeepSeek-V3 94% → 76% from literal to semantic POIs, exposing a critical grounding gap.

Neuro-Symbolic Planning

- **Stage I · NL2DSL translation:** LLM + Reflexion + DSL syntax checker translate open-ended queries into DSL code over up to 5 iterations.
- **Stage II · Interactive search:** A symbolic sketch orchestrates activity selection step-by-step, guided by LLM-ranked POI recommendations; DSL validates each step and **backtracks on violation** (5-min limit/query).



- **Clean decomposition:** understanding (NL) · planning (DSL search) · acting (API tool calls) scales to long-horizon itineraries pure-text generation cannot.
- **Compared to prior NeSy pipelines:** TTG (MILP, ~600K constraints/2-day) becomes infeasible; LLM-modulo's verifier-feedback loop stalls after 3–5 refinement rounds.

Main Results | Rigorous, Cheap NeSy Solution

- **NeSy Planning (DeepSeek-V3) dominates** across Easy / Human-Val / Human-Test: **52.6 / 37.0 / 23.3 FPR**.
- Pure-LLM baselines collapse to **≤6% FPR** despite >94% Delivery Rate.
- Robust under C-LPR, a stricter metric that rejects env-shortcut shortcuts (e.g., mis-reporting cost to pass budget).
- NeSy Planning costs only $0.31 / query with GPT-4o, ~3.6× cheaper than LLM-modulo, ~7.8× cheaper than ReAct.

		DR	EPR Mic.	EPR Mac.	LPR Mic.	LPR Mac.	C-LPR FPR	DR	EPR Mic.	EPR Mac.	LPR Mic.	LPR Mac.	C-LPR FPR
		Easy (#300)						Human-Val (#154)					
Act		70.4	49.9	0	64.6	30.6	0	0			-		
		97.5	70.8	0	86.8	68.6	0	0			-		
ReAct (zero-shot)		43.3	40.8	0	41.9	19.6	0	0	36.4	29.5	0.65	35.2	16.2	0.38	0
		95.4	48.2	0	71.3	33.0	0	0	**96.1**	50.5	0	**72.4**	32.5	0	0
ReAct (one-shot)		77.5	68.3	6.00	74.1	52.3	5.77	5.33	55.2	**57.3**	2.59	64.6	44.2	1.71	2.59
		94.2	68.1	0	**89.4**	**70.6**	0	0	69.5	46.3	0	63.6	46.8	0	0
NeSy Planning		75.3	**75.3**	75.3	70.4	52.6	70.4	52.6	51.9	53.2	52.5	47.0	**37.6**	46.5	**37.0**
		75.0	73.6	**64.0**	73.5	63.3	**61.7**	**60.6**	45.4	50.1	45.4	40.9	29.8	**38.5**	27.9
		72.3	67.0	34.0	70.4	49.6	32.6	28.3	42.8	47.4	42.2	36.2	27.2	34.4	25.3
		32.0	31.9	31.3	29.1	21.0	28.3	21.0	25.9	25.8	24.0	22.3	12.3	20.5	11.0
		30.3	30.3	30.3	27.6	19.6	27.6	19.6	37.6	38.2	37.6	32.7	18.8	32.2	18.8
TTG (oracle)		18.3	21.5	8.66	17.2	15.0	8.23	8.66	9.09	12.8	2.59	7.65	5.19	2.39	1.29
LLM-Modulo* (Oracle Verifier)		48.3	94.5	4.33	58.4	43.6	4.11	4.33	61.6	90.2	2.59	75.9	51.2	2.75	2.59
		91.6	88.2	7.66	**95.5**	**84.6**	7.66	7.00	91.5	87.2	3.24	**92.9**	**66.2**	2.87	3.24
		30.0	80.5	0.0	62.7	25.0	0.0	0.0	35.0	75.3	0.0	61.6	19.4	0.0	0.0
		28.6	69.4	0.0	55.2	8.33	0.0	0.0	19.4	74.1	0.0	43.4	5.19	0.0	0.0
		10.3	90.5	0.0	39.1	9.0	0.0	0.0	3.24	**92.2**	0.0	31.4	4.54	0.0	0.0
NeSy Planning* (Oracle Translation)		**82.6**	**81.7**	**75.0**	**82.2**	75.3	**75.0**	**74.0**	**58.4**	59.6	**57.7**	53.8	46.1	**52.0**	**45.4**
		66.6	66.7	66.0	64.6	63.6	64.6	62.6	52.6	46.9	42.9	47.6	40.9	43.9	40.9
		69.3	69.3	59.3	70.2	59.6	59.3	57.9	53.2	55.1	54.5	48.0	42.8	47.6	40.9
		52.6	52.6	52.6	50.4	45.3	50.4	45.6	40.9	42.8	42.8	37.7	28.5	37.7	27.9
		33.3	33.2	32.6	32.1	32.0	31.4	32.3	29.2	29.1	26.6	25.4	20.1	23.4	19.4
		Human-Test (#1000)						NeSy Planning* (Oracle Translation)					
NeSy Planning		**44.6**	**44.5**	**42.6**	**38.7**	**23.3**	**37.6**	**23.3**	**60.6**	**60.3**	**59.0**	**53.6**	**32.0**	**52.5**	**31.6**
		37.3	37.2	35.0	30.7	11.3	29.2	11.3	27.8	27.8	27.1	24.8	12.8	24.4	12.8
		36.6	36.5	34.6	29.6	6.43	28.5	6.43	41.1	41.1	40.6	34.6	13.8	34.2	13.8

Cost per query

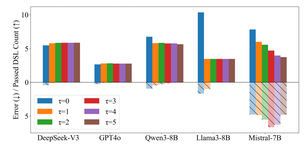
Method	#Input	#Output	($)	($)
Act	88K	2K	007	.219
ReAct (0-shot)	206K	3K	.021	.638
ReAct (1-shot)	1058K	4K	.081	2.43
LLM-modulo	362K	11K	.025	1.12
NeSy Planning	467K	3K	.003	.306

Open-World Challenges Exposed

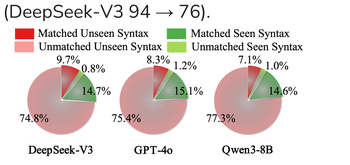
DSL Syntax Compliance. LLMs trade syntax errors for dropped constraints: iterative refinement (e.g., Reflexion) cuts parser errors but under-specifies plans; GPT-4o emits ~2 fewer constraints per query than DeepSeek-V3.

Unseen Compositions. 84% of Human-1000 contains novel DSL structures; LLMs align only 12% on them versus 93% on seen, leaving compositional generalization open.

Contextual Grounding. 78.4% of human DSL needs semantic POI inference; LLMs drop sharply from literal to semantic (DeepSeek-V3 94 → 76).



Syntax error of DSL Translation



Seen vs unseen DSL.

Take-Home Messages

- **Compositional generalization: the emerging frontier for language agents.** Natural language instructions compose atomic concepts with logical syntax, so following open-ended requests inherently demands compositional generalization.
- **Neuro-Symbolic: a promising but imperfect path.** Decomposing language, planning, and action outperforms pure LLMs, yet two fronts remain open: compositional NL → DSL grounding, and the runtime cost of symbolic search.
- **ChinaTravel, a research platform beyond travel planning.** A sandbox with verifiable rewards, a DSL for compositional validation, and authentic open-ended human queries together form a general testbed for constraint-aware language agents.