# iTIMO: An LLM-empowered Synthesis Dataset for Travel Itinerary Modification

Zhuoxuan Huang
lilhzx@csu.edu.cn
Central South University
Changsha, Hunan, China

Yunshan Ma
ysma@smu.edu.sg
Singapore Management University
Singapore, Singapore

Hongyu Zhang*
hyzhang@csu.edu.cn
Central South University
Changsha, Hunan, China

Hua Ma
huama@hunnu.edu.cn
Hunan Normal University
Changsha, Hunan, China

Zhu Sun*
zhu_sun@sutd.edu.sg
Singapore University of Technology and Design
Singapore, Singapore

## 1 Appendix

### 1.1 Dataset Preprocessing

As previously mentioned, our iTIMO datasets are constructed from public travel datasets covering Toronto [1], Melbourne [3], and Florence [2]. We preprocess these datasets as follows. First, itineraries with lengths less than 3 are removed to enable the execution of the DELETE perturbation operation. Second, the Florence dataset contains several itineraries with excessive lengths (*e.g.,* 164 POIs), which are impractical for realistic planning. This issue arises from inconsistent definitions of an itinerary during data collection. Specifically, itineraries in the Melbourne and Toronto datasets are constrained by a total duration of less than 10 hours, whereas the Florence dataset only requires the interval visiting time between adjacent POIs to be less than 6 hours. To address this, we use a sliding window to truncate long itineraries in the Florence dataset. The window size is set to 21, consistent with the maximum itinerary length observed in the Melbourne and Toronto datasets.

**Table 1: Training hyperparameters for SFT LLMs.**

| Hyperparameter | Gemma3 | | Llama3 | | Qwen3 | |
|---|---|---|---|---|---|---|
| | LoRA | Full | LoRA | Full | LoRA | Full |
| epoch | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 |
| batch_size | 4 | 4 | 4 | 4 | 4 | 4 |
| learning_rate | $5\times10^{-5}$ | $1\times10^{-5}$ | $5\times10^{-5}$ | $1\times10^{-5}$ | $5\times10^{-5}$ | $1\times10^{-5}$ |
| weight_decay | 0.01 | 0.00 | 0.01 | 0.01 | 0.01 | 0.01 |
| warmup_step | 30 | 30 | 30 | 30 | 30 | 30 |
| lr_scheduler | cosine | cosine | linear | linear | linear | linear |
| max_grad_norm | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |

### 1.2 Implementation Details

For these base LLMs, we set the temperature to 0 for fixed output. Regarding the LRMs, their temperatures are not allowed to adjust and we use their default settings. In terms of the SFT LLMs, we set the maximum of context length to 8192 and AdamW is selected as the optimizer. For the remaining parameters, grid search is conducted to find the optimal settings. Specifically, the training epoch is selected from $[1.0, 2.0, 3.0]$, batch size is selected from $[4, 8, 16]$, learning rate is selected from $[1\times10^{-6}, 2\times10^{-6}, 5\times10^{-6}, 1\times10^{-5}, 5\times10^{-5}]$, weight decay is selected from $[0.0, 0.01]$, warm-up step is selected from $[10, 30]$, learning rate scheduler is selected from $[cosine, linear]$, and the threshold for gradient clip is selected from $[0.5, 1.0]$. Due to the substantial computational overhead of grid search, we prioritize the validation set of the iTIMO-Melbourne dataset for hyperparameter tuning. Specifically, we focus on the ADD operation, which represents the most challenging itinerary modification task. As a result, the hyperparameter settings are shown in Table 1.

**Table 2: Tool box of itinerary perturbation for our V3.2 FM.**

| Function | Purpose |
|---|---|
| `geo_distance_segments` | Computes consecutive spatial distances between adjacent POIs and discretizes each segment into $low/medium/high$ classes for original and perturbed itineraries. |
| `stats_from_categories` | Given categorical label sequences (e.g., popularity labels or spatial segment classes) of original and perturbed itineraries, computes distribution shift (Hellinger) and rank stability (Kendall's Tau-b), then returns a thresholded disruption decision. |
| `cd_from_categories` | Computes category diversity (CD) of original and perturbed itineraries using case-insensitive unique categories and outputs whether the CD ratio changes (CD disruption). |
| `categories_from_itinerary` | Extracts the POI category from each itinerary to avoid free-text parsing. |

### 1.3 Tool Box for our V3.2 FM

The tool box used in our V3.2 FM is introduced in Table 2.

### 1.4 Detailed Results for RQ1 & RQ4

The detailed results about LLM's performance in the itinerary modification task are displayed in Table 3 and Table 4, respectively.

*Corresponding Authors.

## 1.5 Prompt Template

For the baseline methods (*i.e.,* DeepSeek V3.2 and R3.2), the prompts to guide them for itinerary perturbation are shown as box 1-3. For our proposed V3.2 FM, the perturbation prompts are shown as box 4-6. The template of memory module is shown as box 7. Furthermore, the prompts used for itinerary modification are shown as box 8-10.

**Table 3: Itinerary modification performance across datasets, operations, LLMs, and RAG Settings, where 🟦 denotes LLM, 🟪 denotes LRM, 🟧 denotes LLMs with SFT, `bold` scores indicate the best performance in each column, and `underlined` scores denote the optimal results within specific models.**

| Methods | iTIMO-Melbourne | | | | | | iTIMO-Toronto | | | | | | iTIMO-Florence | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADD | | REPLACE | | DELETE | | ADD | | REPLACE | | DELETE | | ADD | | REPLACE | | DELETE | |
| | Mod↑ | APR↑ | Mod↑ | APR↑ | Mod↑ | APR↑ | Mod↑ | APR↑ | Mod↑ | APR↑ | Mod↑ | APR↑ | Mod↑ | APR↑ | Mod↑ | APR↑ | Mod↑ | APR↑ |
| Gemma3-4b | 0.0921 | 0.2237 | 0.1605 | 0.1235 | 0.3333 | 0.3951 | 0.0769 | 0.2154 | 0.1194 | 0.1493 | 0.1176 | 0.3088 | 0.0723 | 0.3431 | 0.0992 | 0.1277 | 0.1507 | 0.2486 |
| +Few-shot | 0.1447 | 0.3947 | 0.1111 | 0.1481 | 0.2963 | 0.3580 | 0.0615 | 0.2615 | 0.1194 | 0.1642 | 0.3529 | 0.4412 | 0.0502 | 0.2882 | 0.0775 | 0.1060 | 0.1552 | 0.2475 |
| +RAG (Hint) | 0.1316 | 0.2895 | 0.1481 | 0.2099 | 0.2469 | 0.3333 | 0.0923 | 0.3692 | 0.0149 | 0.1642 | 0.2794 | 0.3382 | 0.0548 | 0.3092 | 0.0878 | 0.1231 | 0.1462 | 0.2508 |
| +RAG (GPT-text-emd-3-large) | 0.1184 | 0.3421 | 0.1235 | 0.1728 | 0.2840 | 0.3580 | 0.1231 | 0.3385 | 0.1642 | 0.2090 | 0.3529 | 0.4559 | 0.0735 | 0.3594 | 0.0969 | 0.1300 | 0.1620 | 0.2801 |
| +RAG (KaLM-Gemma3-12b-emd) | 0.0921 | 0.2895 | 0.1358 | 0.1481 | 0.3457 | 0.3951 | 0.1077 | 0.3077 | 0.1343 | 0.1791 | 0.3676 | 0.4853 | 0.0817 | 0.3466 | 0.0981 | 0.1345 | 0.1462 | 0.2565 |
| +RAG (Qwen3-8b-emd) | 0.1316 | 0.4211 | 0.1605 | 0.2099 | 0.3580 | 0.4691 | 0.1385 | 0.3692 | 0.1343 | 0.1940 | 0.3971 | 0.5000 | 0.0735 | 0.3582 | 0.0787 | 0.1254 | 0.1519 | 0.2531 |
| Llama-3.1-8b-Instruct | 0.0921 | 0.2895 | 0.0741 | 0.1358 | 0.2346 | 0.2840 | 0.0769 | 0.3077 | 0.0746 | 0.1045 | 0.1765 | 0.3676 | 0.0572 | 0.3081 | 0.0730 | 0.1288 | 0.1834 | 0.2598 |
| +Few-shot | 0.0789 | 0.2895 | 0.0625 | 0.0988 | 0.2963 | 0.3210 | 0.0000 | 0.3077 | 0.1045 | 0.1194 | 0.2353 | 0.4265 | 0.0642 | 0.3232 | 0.0753 | 0.1277 | 0.2092 | 0.2857 |
| +RAG (Hint) | 0.0789 | 0.3026 | 0.0617 | 0.0988 | 0.2840 | 0.3333 | 0.0462 | 0.2615 | 0.0448 | 0.0597 | 0.2059 | 0.4265 | 0.0782 | 0.3652 | 0.0604 | 0.1129 | 0.2238 | 0.2958 |
| +RAG (GPT-text-emd-3-large) | 0.0658 | 0.2632 | 0.0494 | 0.1111 | 0.3210 | 0.3580 | 0.0923 | 0.2615 | 0.0896 | 0.1343 | 0.2500 | 0.4118 | 0.0770 | 0.3431 | 0.0719 | 0.1300 | 0.2385 | 0.3093 |
| +RAG (KaLM-Gemma3-12b-emd) | 0.1184 | 0.3158 | 0.0617 | 0.0988 | 0.3580 | 0.3827 | 0.1077 | 0.3077 | 0.1045 | 0.1343 | 0.2353 | 0.4265 | 0.0665 | 0.3559 | 0.1129 | 0.1129 | 0.2160 | 0.3003 |
| +RAG (Qwen3-8b-emd) | 0.0526 | 0.2500 | 0.0494 | 0.0864 | 0.2963 | 0.3457 | 0.0923 | 0.3231 | 0.1343 | 0.1940 | 0.2941 | 0.4412 | 0.0758 | 0.3489 | 0.0753 | 0.1243 | 0.2486 | 0.3228 |
| Qwen3-8b | 0.1184 | 0.3816 | 0.0247 | 0.0494 | 0.3457 | 0.3704 | 0.0769 | 0.3846 | 0.1194 | 0.1940 | 0.3235 | 0.5147 | 0.0852 | 0.3967 | 0.1129 | 0.1653 | 0.2418 | 0.3285 |
| +Few-shot | 0.1316 | 0.3947 | 0.0988 | 0.1728 | 0.4691 | 0.5185 | 0.0769 | 0.2769 | 0.0896 | 0.1493 | 0.4853 | 0.5441 | 0.0933 | 0.4107 | 0.1380 | 0.1893 | 0.2250 | 0.3116 |
| +RAG (Hint) | 0.1447 | 0.4342 | 0.1111 | 0.1852 | 0.4198 | 0.4815 | 0.1231 | 0.4000 | 0.1343 | 0.1940 | 0.4265 | 0.5147 | 0.1050 | 0.4492 | 0.1494 | 0.1973 | 0.2306 | 0.3251 |
| +RAG (GPT-text-emd-3-large) | 0.1579 | 0.4079 | 0.1111 | 0.1975 | 0.4691 | 0.5185 | 0.1077 | 0.3231 | 0.1642 | 0.2239 | 0.4559 | 0.5588 | 0.1027 | 0.4376 | 0.1391 | 0.1938 | 0.2553 | 0.3442 |
| +RAG (KaLM-Gemma3-12b-emd) | 0.1447 | 0.4211 | 0.1235 | 0.1728 | 0.4321 | 0.4938 | 0.0923 | 0.3231 | 0.0896 | 0.1642 | 0.4853 | 0.5882 | 0.1190 | 0.4597 | 0.1448 | 0.2052 | 0.2407 | 0.3251 |
| +RAG (Qwen3-8b-emd) | 0.0921 | 0.3816 | 0.1111 | 0.1605 | 0.3827 | 0.4444 | 0.0923 | 0.3385 | 0.1940 | 0.2388 | 0.4853 | 0.5882 | 0.0992 | 0.4387 | 0.1345 | 0.1881 | 0.2160 | 0.3161 |
| Phi4-15b | 0.1316 | 0.3553 | 0.0864 | 0.1235 | 0.1852 | 0.1852 | 0.0308 | 0.1692 | 0.0597 | 0.1343 | 0.1176 | 0.3382 | 0.0548 | 0.3197 | 0.0616 | 0.1266 | 0.1654 | 0.2441 |
| +Few-shot | 0.0789 | 0.2895 | 0.1481 | 0.2099 | 0.2716 | 0.2963 | 0.0462 | 0.1692 | 0.0746 | 0.1791 | 0.1618 | 0.3971 | 0.0630 | 0.3606 | 0.0787 | 0.1437 | 0.2171 | 0.2970 |
| +RAG (Hint) | 0.1053 | 0.3026 | 0.0864 | 0.1358 | 0.2593 | 0.3086 | 0.0769 | 0.1692 | 0.0896 | 0.1642 | 0.1765 | 0.3676 | 0.0665 | 0.3512 | 0.0604 | 0.1220 | 0.2013 | 0.2846 |
| +RAG (GPT-text-emd-3-large) | 0.1184 | 0.3158 | 0.1235 | 0.1605 | 0.2593 | 0.2840 | 0.0923 | 0.2154 | 0.0448 | 0.1045 | 0.2353 | 0.4412 | 0.0782 | 0.3676 | 0.0844 | 0.1448 | 0.2351 | 0.3116 |
| +RAG (KaLM-Gemma3-12b-emd) | 0.1184 | 0.3421 | 0.0741 | 0.1235 | 0.2593 | 0.2840 | 0.0308 | 0.1692 | 0.0896 | 0.1791 | 0.2353 | 0.4118 | 0.0630 | 0.3524 | 0.0787 | 0.1391 | 0.2238 | 0.3093 |
| +RAG (Qwen3-8b-emd) | 0.1447 | 0.2895 | 0.0988 | 0.1605 | 0.2716 | 0.2963 | 0.0769 | 0.2923 | 0.0746 | 0.1642 | 0.3235 | 0.5294 | 0.0712 | 0.3594 | 0.0844 | 0.1448 | 0.2272 | 0.3071 |
| Mistral-24b | 0.1184 | 0.3289 | 0.1728 | 0.2222 | 0.3210 | 0.3457 | 0.0462 | 0.2308 | 0.0896 | 0.1194 | 0.1618 | 0.3676 | 0.0723 | 0.3547 | 0.0867 | 0.1437 | 0.1766 | 0.2497 |
| +Few-shot | 0.1184 | 0.3684 | 0.1358 | 0.1975 | 0.4444 | 0.4568 | 0.0923 | 0.3077 | 0.0746 | 0.1194 | 0.2353 | 0.4412 | 0.0817 | 0.4037 | 0.1243 | 0.1813 | 0.2610 | 0.3510 |
| +RAG (Hint) | 0.1316 | 0.3553 | 0.1605 | 0.2593 | 0.4444 | 0.4815 | 0.0923 | 0.3692 | 0.1045 | 0.1791 | 0.3529 | 0.5441 | 0.1062 | 0.4096 | 0.1414 | 0.1961 | 0.2688 | 0.3555 |
| +RAG (GPT-text-emd-3-large) | 0.1842 | 0.4868 | 0.1852 | 0.2469 | 0.4691 | 0.5062 | 0.1385 | 0.4154 | 0.1791 | 0.2090 | 0.3382 | 0.5294 | 0.1214 | 0.4807 | 0.1688 | 0.2281 | 0.3251 | 0.4128 |
| +RAG (KaLM-Gemma3-12b-emd) | 0.1316 | 0.3816 | 0.1605 | 0.2469 | 0.4691 | 0.5062 | 0.1077 | 0.3846 | 0.2388 | 0.3134 | 0.3824 | 0.5000 | 0.1179 | 0.4574 | 0.1551 | 0.2121 | 0.3116 | 0.3870 |
| +RAG (Qwen3-8b-emd) | 0.1579 | 0.4079 | 0.1481 | 0.2222 | 0.4815 | 0.5185 | 0.1385 | 0.4154 | 0.1493 | 0.2239 | 0.3382 | 0.5000 | 0.1085 | 0.4469 | 0.1517 | 0.2269 | 0.3048 | 0.3847 |
| Qwen3-32b | 0.1711 | 0.3947 | 0.1235 | 0.1852 | 0.3457 | 0.3951 | 0.0615 | 0.2154 | 0.1194 | 0.1493 | 0.2500 | 0.3529 | 0.0654 | 0.3419 | 0.0833 | 0.1311 | 0.2418 | 0.3318 |
| +Few-shot | 0.2368 | 0.5132 | 0.1481 | 0.1975 | 0.4815 | 0.5679 | 0.0923 | 0.3846 | 0.1343 | 0.1791 | 0.4118 | 0.5441 | 0.0793 | 0.4026 | 0.1106 | 0.1688 | 0.3273 | 0.4162 |
| +RAG (Hint) | 0.1579 | 0.5000 | 0.1852 | 0.2469 | 0.3951 | 0.4815 | 0.1231 | 0.1231 | 0.1493 | 0.2239 | 0.4118 | 0.5147 | 0.1029 | 0.4247 | 0.1209 | 0.1688 | 0.2981 | 0.3847 |
| +RAG (GPT-text-emd-3-large) | 0.1579 | 0.4605 | 0.1852 | 0.1975 | 0.3704 | 0.4691 | 0.1692 | 0.4154 | 0.1940 | 0.2985 | 0.4559 | 0.5735 | 0.1121 | 0.4492 | 0.1551 | 0.2018 | 0.3251 | 0.4061 |
| +RAG (KaLM-Gemma3-12b-emd) | 0.1711 | 0.5526 | 0.2099 | 0.2222 | 0.3827 | 0.4568 | 0.1077 | 0.4615 | 0.2090 | 0.2687 | 0.4559 | 0.5441 | 0.1180 | 0.4551 | 0.1631 | 0.2189 | 0.3240 | 0.4184 |
| +RAG (Qwen3-8b-emd) | 0.1316 | 0.4342 | 0.1728 | 0.2099 | 0.3827 | 0.4568 | 0.0923 | 0.4308 | 0.1045 | 0.2239 | 0.3676 | 0.4853 | 0.1155 | 0.4481 | 0.1574 | 0.2166 | 0.3566 | 0.4376 |
| GPT-4.1 | 0.0789 | 0.2895 | 0.1111 | 0.1728 | 0.4691 | 0.5062 | 0.0462 | 0.1385 | 0.0896 | 0.1493 | 0.5882 | 0.7500 | 0.0747 | 0.3221 | 0.0662 | 0.1323 | 0.4162 | 0.4858 |
| +Few-shot | 0.1312 | 0.2763 | 0.1111 | 0.1728 | 0.5556 | 0.6173 | 0.0615 | 0.2154 | 0.0896 | 0.1791 | 0.6176 | 0.7500 | 0.0712 | 0.3092 | 0.0753 | 0.1437 | 0.3813 | 0.4612 |
| +RAG (Hint) | 0.0658 | 0.2632 | 0.1235 | 0.1852 | 0.4568 | 0.4815 | 0.0615 | 0.2000 | 0.1493 | 0.2239 | 0.5441 | 0.6912 | 0.0828 | 0.3174 | 0.0696 | 0.1345 | 0.2767 | 0.3453 |
| +RAG (GPT-text-emd-3-large) | 0.0526 | 0.2763 | 0.0617 | 0.1605 | 0.4198 | 0.4568 | 0.1077 | 0.2769 | 0.1642 | 0.2090 | 0.5441 | 0.7206 | 0.0770 | 0.3209 | 0.0993 | 0.1642 | 0.3127 | 0.3903 |
| +RAG (KaLM-Gemma3-12b-emd) | 0.0921 | 0.3289 | 0.1111 | 0.2222 | 0.4074 | 0.4321 | 0.0923 | 0.2000 | 0.1194 | 0.2090 | 0.6176 | 0.7353 | 0.0758 | 0.3361 | 0.0845 | 0.1505 | 0.3183 | 0.3937 |
| +RAG (Qwen3-8b-emd) | 0.1316 | 0.3947 | 0.0988 | 0.1481 | 0.4444 | 0.4691 | 0.1077 | 0.2923 | 0.1791 | 0.2537 | 0.5441 | 0.6765 | 0.0782 | 0.3314 | 0.0959 | 0.1608 | 0.3746 | 0.4409 |
| DeepSeek-V3.2 | 0.0658 | 0.3026 | 0.1358 | 0.1728 | 0.6173 | 0.7037 | 0.0154 | 0.0615 | 0.0746 | 0.1194 | 0.5882 | 0.7059 | 0.0665 | 0.2964 | 0.0901 | 0.1494 | 0.3746 | 0.4477 |
| +Few-shot | 0.0789 | 0.2500 | 0.0988 | 0.1605 | 0.5802 | 0.6543 | 0.0308 | 0.1077 | 0.1343 | 0.1343 | 0.6029 | 0.6912 | 0.0770 | 0.3372 | 0.1026 | 0.1585 | 0.4466 | 0.5354 |
| +RAG (Hint) | 0.0789 | 0.3026 | 0.1111 | 0.1605 | 0.5802 | 0.6667 | 0.0308 | 0.1077 | 0.0896 | 0.1343 | 0.5441 | 0.5735 | 0.0712 | 0.3326 | 0.0981 | 0.1642 | 0.4623 | 0.5399 |
| +RAG (GPT-text-emd-3-large) | 0.0789 | 0.2500 | 0.1481 | 0.1728 | 0.5926 | 0.6667 | 0.0769 | 0.1846 | 0.1493 | 0.1493 | 0.6324 | 0.6912 | 0.0910 | 0.3547 | 0.1357 | 0.1916 | 0.4994 | 0.5759 |
| +RAG (KaLM-Gemma3-12b-emd) | 0.1184 | 0.2895 | 0.1235 | 0.1605 | 0.6173 | 0.6914 | 0.0000 | 0.1077 | 0.1045 | 0.1343 | 0.6618 | 0.7353 | 0.0898 | 0.3489 | 0.1186 | 0.1767 | 0.4747 | 0.5624 |
| +RAG (Qwen3-8b-emd) | 0.0921 | 0.2632 | 0.1728 | 0.2222 | 0.6269 | 0.7037 | 0.0769 | 0.1692 | 0.1642 | 0.1642 | 0.6029 | 0.6618 | 0.0723 | 0.3419 | 0.1471 | 0.2166 | 0.5242 | 0.5951 |
| GPT-o4-mini | 0.1974 | 0.6184 | 0.3333 | 0.5432 | 0.5679 | 0.6543 | 0.3385 | 0.6923 | 0.3433 | 0.5672 | 0.6471 | 0.7941 | 0.2614 | 0.6966 | 0.3627 | 0.4036 | 0.5130 | 0.6142 |
| +Few-shot | 0.2105 | 0.6711 | 0.3210 | 0.5062 | 0.6049 | 0.6543 | 0.3538 | 0.7231 | 0.3731 | 0.5672 | 0.6029 | 0.7500 | 0.2515 | 0.6919 | 0.3879 | 0.4242 | 0.5107 | 0.6198 |
| DeepSeek-R3.2 | **0.2763** | 0.6842 | 0.3457 | 0.5309 | 0.5926 | 0.6543 | 0.2462 | **0.7538** | 0.3284 | 0.4925 | 0.6029 | 0.7353 | 0.1715 | 0.7083 | 0.3307 | 0.4835 | 0.5259 | 0.6232 |
| +Few-shot | 0.2632 | **0.6974** | **0.3951** | **0.6296** | 0.5875 | 0.6296 | **0.2615** | 0.7385 | 0.3284 | 0.5075 | 0.5735 | 0.7500 | 0.1688 | 0.6919 | 0.3330 | 0.4789 | 0.5181 | 0.6232 |
| Gemma3-4b (+LoRA) | 0.2368 | 0.5132 | 0.3333 | 0.4198 | 0.4321 | 0.4568 | 0.2000 | 0.6000 | 0.2985 | 0.3731 | 0.6029 | 0.6765 | 0.2275 | 0.6418 | 0.4367 | 0.5108 | 0.6805 | 0.7514 |
| +Few-shot | 0.1184 | 0.4079 | 0.1481 | 0.1728 | 0.2963 | 0.3333 | 0.1231 | 0.3538 | 0.0896 | 0.1791 | 0.3971 | 0.4559 | 0.1610 | 0.5228 | 0.2155 | 0.2759 | 0.4094 | 0.5051 |
| +RAG (Hint) | 0.1184 | 0.3421 | 0.1975 | 0.2346 | 0.2469 | 0.3457 | 0.2154 | 0.5077 | 0.1045 | 0.1493 | 0.4559 | 0.5000 | 0.1622 | 0.5321 | 0.2463 | 0.3044 | 0.4477 | 0.5264 |
| +RAG (GPT-text-emd-3-large) | 0.1842 | 0.3947 | 0.1728 | 0.1852 | 0.2963 | 0.3704 | 0.1538 | 0.4462 | 0.1940 | 0.2985 | 0.3235 | 0.4118 | 0.1645 | 0.5309 | 0.2372 | 0.3067 | 0.4184 | 0.5129 |
| +RAG (KaLM-Gemma3-12b-emd) | 0.1316 | 0.3553 | 0.1235 | 0.1852 | 0.2840 | 0.3333 | 0.1077 | 0.4462 | 0.2239 | 0.2687 | 0.4265 | 0.4853 | 0.1680 | 0.5554 | 0.2383 | 0.2987 | 0.4218 | 0.5051 |
| +RAG (Qwen3-8b-emd) | 0.1447 | 0.3816 | 0.1358 | 0.1728 | 0.2840 | 0.4321 | 0.1692 | 0.4769 | 0.2388 | 0.3284 | 0.3088 | 0.3971 | 0.1657 | 0.5438 | 0.1881 | 0.2486 | 0.3532 | 0.4567 |
| LLama3.1-8b-Instruct (+LoRA) | 0.2500 | 0.6053 | 0.3827 | 0.4321 | **0.6914** | **0.7160** | 0.2154 | 0.6462 | 0.4627 | 0.5224 | **0.7647** | **0.8529** | 0.2695 | 0.6779 | 0.5735 | 0.6420 | 0.7300 | 0.7874 |
| +Few-shot | 0.2632 | 0.6447 | 0.3333 | 0.3951 | 0.6049 | 0.6420 | **0.2615** | 0.6154 | 0.3134 | 0.4030 | 0.6912 | 0.7647 | 0.2299 | 0.6336 | 0.4732 | 0.5678 | 0.7199 | 0.7705 |
| +RAG (Hint) | 0.1974 | 0.5789 | **0.3827** | 0.4938 | 0.5802 | 0.6296 | 0.1385 | 0.5538 | 0.3582 | 0.4328 | 0.6618 | 0.7353 | 0.2299 | 0.6441 | 0.5154 | 0.5986 | 0.7132 | 0.7683 |
| +RAG (GPT-text-emd-3-large) | 0.2368 | 0.5789 | 0.3333 | 0.3951 | 0.5926 | 0.6420 | 0.2308 | 0.6000 | 0.3881 | 0.4776 | 0.6618 | 0.7206 | 0.2485 | 0.6698 | 0.5017 | 0.5838 | 0.7120 | 0.7672 |
| +RAG (KaLM-Gemma3-12b-emd) | 0.2105 | 0.5526 | 0.2593 | 0.3704 | 0.5926 | 0.6296 | 0.2308 | 0.6462 | 0.4627 | 0.5224 | 0.7206 | 0.7794 | 0.2357 | 0.6441 | 0.4903 | 0.5667 | 0.7087 | 0.7548 |
| +RAG (Qwen3-8b-emd) | 0.2500 | 0.5658 | 0.3580 | 0.4198 | 0.5185 | 0.5679 | 0.2000 | 0.6308 | 0.4328 | 0.5075 | 0.7500 | 0.8088 | 0.2544 | 0.6756 | 0.5017 | 0.5792 | 0.6929 | 0.7570 |
| Qwen3-8b (+LoRA) | 0.2632 | 0.6316 | 0.3580 | 0.4444 | 0.5309 | 0.5556 | 0.1846 | 0.5846 | 0.3134 | 0.4179 | 0.6471 | 0.7059 | 0.2602 | 0.6896 | 0.5633 | 0.6488 | 0.7098 | 0.7683 |
| +Few-shot | 0.1842 | 0.6053 | 0.2346 | 0.2840 | 0.5556 | 0.5926 | 0.1231 | 0.5077 | 0.2537 | 0.3284 | 0.6324 | 0.7059 | 0.2065 | 0.6289 | 0.4971 | 0.5872 | 0.6535 | 0.7199 |
| +RAG (Hint) | **0.3026** | 0.6184 | 0.3210 | 0.4198 | 0.5802 | 0.6420 | 0.1385 | 0.4923 | 0.3881 | 0.4328 | 0.6324 | 0.7059 | 0.1995 | 0.6231 | 0.4914 | 0.5804 | 0.6682 | 0.7244 |
| +RAG (GPT-text-emd-3-large) | 0.2500 | 0.5921 | 0.3951 | 0.4321 | 0.5679 | 0.6296 | 0.1538 | 0.5231 | 0.3433 | 0.3881 | 0.6176 | 0.6765 | 0.2217 | 0.6418 | 0.5006 | 0.5838 | 0.6398 | 0.6704 |
| +RAG (KaLM-Gemma3-12b-emd) | 0.1974 | 0.5526 | 0.3457 | 0.3827 | 0.5432 | 0.6173 | 0.1077 | 0.5538 | 0.4030 | 0.4776 | 0.6471 | 0.7500 | 0.2124 | 0.6441 | 0.4983 | 0.5895 | 0.6457 | 0.6502 |
| +RAG (Qwen3-8b-emd) | 0.2237 | 0.5789 | 0.3210 | 0.4074 | 0.5926 | 0.6420 | 0.1538 | 0.5077 | 0.3881 | 0.4478 | 0.6176 | 0.6912 | 0.2229 | 0.6313 | 0.5086 | 0.5770 | 0.6575 | 0.7087 |

**Table 4: Effects of various RAG settings on LLMs with SFT, where Δ denotes the relative performance improvement over the zero-shot inference baseline for the same model. Cells shaded with 🟩 indicate $\Delta > 0$ (improvement), and cells shaded with 🟥 indicate $\Delta \leq 0$ (no improvement).**

| Backbone | Training Settings | RAG Settings (Inference) | iTIMO-Melbourne | | | | | | iTIMO-Toronto | | | | | | iTIMO-Florence | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ADD | | REPLACE | | DELETE | | ADD | | REPLACE | | DELETE | | ADD | | REPLACE | | DELETE | |
| | | | $\Delta_{\text{Mod}}$ | $\Delta_{\text{APR}}$ | $\Delta_{\text{Mod}}$ | $\Delta_{\text{APR}}$ | $\Delta_{\text{Mod}}$ | $\Delta_{\text{APR}}$ | $\Delta_{\text{Mod}}$ | $\Delta_{\text{APR}}$ | $\Delta_{\text{Mod}}$ | $\Delta_{\text{APR}}$ | $\Delta_{\text{Mod}}$ | $\Delta_{\text{APR}}$ | $\Delta_{\text{Mod}}$ | $\Delta_{\text{APR}}$ | $\Delta_{\text{Mod}}$ | $\Delta_{\text{APR}}$ | $\Delta_{\text{Mod}}$ | $\Delta_{\text{APR}}$ |
| Gemma3-4b (LoRA) | Zero-shot | Few-shot | -50.0% | -20.5% | -55.6% | -58.8% | -31.4% | -27.0% | -38.5% | -41.0% | -70.0% | -52.0% | -34.1% | -32.6% | -29.2% | -18.5% | -50.6% | -46.0% | -39.8% | -32.8% |
| | | Hint | -50.0% | -33.4% | -40.7% | -44.1% | -42.9% | -24.3% | +7.7% | -15.4% | -65.0% | -60.0% | -24.4% | -26.1% | -28.7% | -17.1% | -43.6% | -40.4% | -34.2% | -29.9% |
| | | GPT-text-emd-large | -22.2% | -23.1% | -48.2% | -55.9% | -31.4% | -18.9% | -23.1% | -25.6% | -35.0% | -20.0% | -46.4% | -39.1% | -27.7% | -17.3% | -45.7% | -39.9% | -38.5% | -31.8% |
| | | KaLM-Gemma3-12b-emd | -44.4% | -30.8% | -63.0% | -55.9% | -34.3% | -27.0% | -46.2% | -25.6% | -25.0% | -28.0% | -29.3% | -28.3% | -26.2% | -13.5% | -45.4% | -41.5% | -38.0% | -32.8% |
| | | Qwen3-8b-emd | -38.9% | -25.6% | -59.3% | -58.8% | -34.3% | -5.4% | -15.4% | -20.5% | -20.0% | -12.0% | -48.8% | -41.3% | -27.2% | -15.3% | -56.9% | -51.3% | -48.1% | -39.2% |
| | Few-shot | Few-shot | +0.0% | +7.7% | -7.4% | -14.7% | +5.7% | +5.4% | +15.4% | -10.3% | +5.0% | +12.0% | -12.2% | -17.4% | +3.4% | +9.5% | +29.2% | +42.4% | +10.6% | +7.8% |
| | | Hint | -11.1% | +0.0% | +1.2% | -17.6% | -17.9% | +0.0% | +0.0% | -3.2% | +1.2% | +8.0% | -12.2% | -17.4% | -2.6% | +9.9% | +20.7% | +35.4% | +11.0% | +7.8% |
| | | GPT-text-emd-large | -22.2% | +0.0% | -7.4% | -11.8% | +2.4% | +2.7% | +23.1% | +6.5% | +11.3% | +8.0% | -7.3% | -4.3% | +4.0% | +8.5% | +26.0% | +40.4% | +10.4% | +9.4% |
| | | KaLM-Gemma3-12b-emd | -5.6% | +0.0% | -2.5% | -14.7% | -5.7% | -2.7% | +23.1% | +6.5% | +26.2% | +24.0% | -2.4% | -4.3% | -3.1% | +5.9% | +27.4% | +42.1% | +10.6% | +7.9% |
| | | Qwen3-8b-emd | -5.6% | +0.0% | +0.0% | -17.6% | -17.9% | +0.0% | +15.4% | +6.5% | +17.5% | +8.0% | -7.3% | -8.7% | +2.9% | +9.1% | +26.6% | +39.6% | +12.1% | +8.9% |
| | Alignment | Hint | +7.1% | -15.4% | +0.0% | +3.8% | -18.9% | -19.0% | -10.0% | -15.6% | -11.1% | -16.0% | +47.8% | +22.2% | +1.7% | +1.0% | +7.8% | +3.6% | -9.5% | -8.2% |
| | | GPT-text-emd-large | +7.1% | -7.7% | -8.3% | -3.8% | +0.0% | -7.1% | +60.0% | +9.4% | -11.1% | -16.0% | +73.9% | +27.8% | +10.1% | +4.1% | +11.8% | +6.8% | -9.6% | -8.0% |
| | | KaLM-Gemma3-12b-emd | -14.3% | -20.5% | +8.3% | +7.7% | -10.8% | -11.9% | +30.0% | +6.2% | +11.1% | +4.0% | +87.0% | +30.6% | +7.8% | +2.1% | +11.2% | +10.4% | -8.6% | -8.3% |
| | | Qwen3-8b-emd | +7.1% | -2.6% | +0.0% | +7.7% | -21.6% | -21.4% | +30.0% | -3.1% | +22.2% | +12.0% | +78.3% | +25.0% | +11.7% | +3.5% | +15.3% | +10.2% | -6.3% | -7.9% |
| Gemma3-4b (FullFT) | Zero-shot | Few-shot | +0.0% | +0.0% | -27.3% | -20.0% | -28.3% | -23.4% | -21.1% | -14.0% | -25.8% | -21.6% | -18.6% | -17.8% | -14.7% | -8.8% | -29.1% | -26.4% | -15.1% | -11.7% |
| | | Hint | +6.7% | +0.0% | -27.3% | -8.0% | -8.7% | -2.1% | -26.3% | +0.0% | -41.9% | -27.0% | -9.3% | -8.9% | -4.3% | -9.4% | -25.8% | -23.7% | -11.4% | -10.3% |
| | | GPT-text-emd-large | +13.3% | +14.0% | -27.3% | -24.0% | -26.1% | -17.0% | -15.8% | -9.3% | -22.6% | -21.6% | -11.6% | -4.4% | -5.9% | -10.3% | -29.4% | -26.7% | -15.8% | -13.6% |
| | | KaLM-Gemma3-12b-emd | +0.0% | -2.3% | -27.3% | -16.0% | -32.6% | -25.5% | -21.1% | +2.3% | -25.8% | -18.9% | -16.3% | -8.9% | -2.0% | -8.1% | -28.6% | -25.2% | -13.5% | -10.8% |
| | | Qwen3-8b-emd | +20.0% | -7.0% | -22.7% | -12.0% | -23.9% | -12.8% | -15.8% | -2.3% | -32.3% | -29.7% | -30.2% | -17.8% | -6.6% | -4.9% | -27.5% | -28.1% | -14.8% | -12.3% |
| | Few-shot | Few-shot | +14.3% | +13.5% | -21.4% | -15.1% | +14.3% | +13.5% | +0.0% | +4.4% | -5.9% | -5.9% | +0.0% | +4.4% | +10.0% | +4.8% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | Hint | +14.3% | +13.5% | -21.4% | -11.3% | +7.1% | +8.1% | +0.0% | +4.4% | -5.9% | -2.9% | -5.3% | -4.4% | +5.0% | +4.8% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | GPT-text-emd-large | +28.6% | +24.3% | -14.3% | -11.3% | +14.3% | +10.8% | +0.0% | +11.1% | -5.9% | -5.9% | +0.0% | +4.4% | +10.0% | +9.5% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | KaLM-Gemma3-12b-emd | +21.4% | +24.3% | -28.6% | -18.9% | +14.3% | +10.8% | +0.0% | +11.1% | -11.8% | -8.8% | +0.0% | +4.4% | +15.0% | +9.5% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | Qwen3-8b-emd | +7.1% | +13.5% | -21.4% | -11.3% | +14.3% | +10.8% | +0.0% | +4.4% | -5.9% | -5.9% | +0.0% | +4.4% | +10.0% | +4.8% | +0.0% | +0.0% | +0.0% | +0.0% |
| Llama3.1-8b (LoRA) | Zero-shot | Few-shot | +5.3% | +6.5% | -12.9% | -8.6% | -12.5% | -10.3% | +21.4% | -4.8% | -32.3% | -22.8% | -9.6% | -10.3% | -14.7% | -6.5% | -17.5% | -11.6% | -1.4% | -2.1% |
| | | Hint | -21.0% | -4.3% | +0.0% | +14.3% | -16.1% | -12.1% | -35.7% | -14.3% | -22.6% | -17.1% | -13.5% | -13.8% | -14.7% | -5.0% | -10.1% | -6.8% | -2.3% | -2.4% |
| | | GPT-text-emd-large | -5.3% | -4.3% | -12.9% | -8.6% | -14.3% | -10.3% | +7.1% | -7.2% | -16.1% | -8.6% | -13.5% | -15.5% | -7.8% | -1.2% | -12.5% | -9.1% | -2.5% | -2.6% |
| | | KaLM-Gemma3-12b-emd | -15.8% | -8.7% | -32.3% | -14.3% | -14.3% | -12.1% | +7.1% | +0.0% | +0.0% | +0.0% | -5.8% | -8.6% | -12.5% | -5.0% | -14.5% | -11.7% | -2.9% | -4.2% |
| | | Qwen3-8b-emd | +0.0% | -6.5% | -6.5% | -2.9% | -25.0% | -20.7% | -7.1% | -2.4% | -6.5% | -2.9% | -1.9% | -5.2% | -5.6% | -0.3% | -12.5% | -9.8% | -5.1% | -3.9% |
| | Few-shot | Few-shot | -12.5% | +15.0% | +6.5% | +25.0% | +0.0% | +12.5% | +0.0% | +0.0% | +8.3% | +11.8% | +0.0% | +8.3% | -2.9% | +27.6% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | Hint | -25.0% | +5.0% | +3.2% | +10.0% | -5.0% | +0.0% | -14.3% | +0.0% | +4.2% | +5.9% | -5.6% | +0.0% | -2.9% | +22.4% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | GPT-text-emd-large | -12.5% | +15.0% | +3.2% | +15.0% | -5.0% | +0.0% | +0.0% | +0.0% | +0.0% | +0.0% | -5.6% | +0.0% | +0.0% | +24.1% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | KaLM-Gemma3-12b-emd | -12.5% | +15.0% | +6.5% | +10.0% | -5.0% | +0.0% | +0.0% | +0.0% | +8.3% | +5.9% | -5.6% | +0.0% | -2.9% | +22.4% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | Qwen3-8b-emd | -12.5% | +15.0% | +3.2% | +10.0% | -5.0% | +0.0% | -14.3% | +0.0% | +0.0% | +0.0% | -5.6% | +0.0% | -2.9% | +22.4% | +0.0% | +0.0% | +0.0% | +0.0% |
| | Alignment | Hint | -15.0% | -6.5% | +13.8% | +33.3% | -22.4% | -15.4% | -30.0% | +11.1% | -3.7% | +6.7% | -6.5% | -4.0% | -22.8% | -3.0% | +3.6% | +2.0% | +1.5% | +1.4% |
| | | GPT-text-emd-large | -25.0% | -6.5% | +3.4% | +20.0% | -8.2% | -13.5% | +70.0% | +16.7% | +7.4% | +16.7% | -2.2% | -2.0% | -10.8% | +2.5% | -4.2% | -3.3% | +4.1% | +3.5% |
| | | KaLM-Gemma3-12b-emd | -5.0% | -10.9% | +20.7% | +33.3% | -16.3% | -15.4% | +20.0% | +5.6% | +14.8% | +10.0% | -10.9% | -10.0% | -14.7% | +0.2% | +1.3% | +0.5% | -0.2% | -1.4% |
| | | Qwen3-8b-emd | -15.0% | -8.7% | +3.4% | +20.0% | +8.2% | +5.8% | +30.0% | +19.4% | -7.4% | +6.7% | +2.2% | +0.0% | +0.9% | +4.2% | +3.8% | +1.6% | -0.3% | +0.8% |
| Llama3.1-8b (FullFT) | Zero-shot | Few-shot | +5.0% | +0.0% | -2.7% | +2.5% | -3.7% | -1.8% | +11.1% | +7.3% | -17.6% | -13.2% | +6.8% | +5.8% | -12.9% | -9.3% | -13.6% | -10.2% | -3.1% | -2.8% |
| | | Hint | -10.0% | -8.3% | -5.4% | +0.0% | -1.9% | -3.5% | -16.7% | -2.4% | -11.8% | -7.9% | +4.5% | +1.9% | -17.2% | -5.9% | -12.2% | -10.4% | -1.8% | -2.1% |
| | | GPT-text-emd-large | -10.0% | -6.3% | -8.1% | +2.5% | -3.7% | -5.3% | -5.6% | +9.8% | -2.9% | +5.3% | +4.5% | +1.9% | -4.2% | -4.7% | -13.4% | -11.1% | -3.7% | -4.4% |
| | | KaLM-Gemma3-12b-emd | -5.0% | -4.2% | -5.4% | +0.0% | +1.9% | +0.0% | +11.1% | +12.2% | -11.8% | -5.3% | +2.3% | +0.0% | -13.2% | -7.1% | -16.3% | -13.8% | -2.0% | -2.8% |
| | | Qwen3-8b-emd | -10.0% | -4.2% | -5.4% | -5.0% | -1.9% | -1.8% | -5.6% | +19.5% | -5.9% | +2.6% | +11.4% | +5.8% | -3.4% | -2.2% | -10.3% | -11.7% | -7.5% | -5.9% |
| | Few-shot | Few-shot | +10.5% | +12.0% | +2.9% | +6.4% | -5.0% | +2.9% | +7.1% | +15.9% | +0.0% | +6.7% | -3.6% | +0.0% | +9.1% | +9.1% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | Hint | +10.5% | +8.0% | +2.9% | +3.2% | -2.5% | +0.0% | +7.1% | +11.4% | -3.7% | +3.3% | -3.6% | -3.8% | +9.1% | +9.1% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | GPT-text-emd-large | +15.8% | +12.0% | +5.7% | +9.6% | -5.0% | +2.9% | +7.1% | +15.9% | -3.7% | +6.7% | -3.6% | -3.8% | +9.1% | +9.1% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | KaLM-Gemma3-12b-emd | +15.8% | +12.0% | +2.9% | +6.4% | -5.0% | +2.9% | +7.1% | +15.9% | -3.7% | +6.7% | -3.6% | +0.0% | +9.1% | +9.1% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | Qwen3-8b-emd | +10.5% | +12.0% | +2.9% | +6.4% | -5.0% | +2.9% | +7.1% | +15.9% | -3.7% | +6.7% | -3.6% | +0.0% | +9.1% | +9.1% | +0.0% | +0.0% | +0.0% | +0.0% |
| Qwen3-8b (LoRA) | Zero-shot | Few-shot | -30.0% | -4.2% | -34.5% | -36.1% | +4.7% | +6.7% | -33.3% | -13.2% | -19.1% | -21.4% | -2.3% | +0.0% | -20.6% | -8.8% | -11.8% | -9.5% | -7.9% | -6.3% |
| | | Hint | +15.0% | -2.1% | -10.3% | -5.6% | +9.3% | +15.6% | -25.0% | -15.8% | +23.8% | +3.6% | -2.3% | +0.0% | -23.3% | -9.6% | -12.8% | -10.5% | -5.9% | -5.7% |
| | | GPT-text-emd-large | -5.0% | -6.2% | +10.4% | -2.8% | +7.0% | +13.3% | -16.7% | -10.5% | +9.5% | -7.1% | -4.6% | -4.2% | -14.8% | -6.9% | -11.1% | -10.0% | -9.9% | -12.7% |
| | | KaLM-Gemma3-12b-emd | -25.0% | -12.5% | -3.4% | -13.9% | +2.3% | +11.1% | -41.7% | -5.3% | +28.6% | +14.3% | +0.0% | +6.3% | -18.4% | -6.6% | -11.5% | -9.2% | -9.0% | -15.4% |
| | | Qwen3-8b-emd | -15.0% | -8.3% | -10.3% | -8.3% | +11.6% | +15.6% | -16.7% | -13.2% | +23.8% | +7.2% | -4.6% | -2.1% | -14.3% | -8.5% | -9.7% | -11.1% | -7.4% | -7.8% |
| | Few-shot | Few-shot | +5.0% | +12.5% | +14.3% | +23.1% | +14.3% | +23.1% | +14.3% | +0.0% | +22.2% | +8.3% | +0.0% | +0.0% | -1.4% | +0.0% | +0.0% | +0.0% | +0.0% | +6.3% |
| | | Hint | +0.0% | +6.2% | +28.6% | +30.8% | +23.8% | +30.8% | +14.3% | +6.0% | +22.2% | +16.7% | +0.0% | +0.0% | -1.4% | +0.0% | +0.0% | +0.0% | +0.0% | +6.3% |
| | | GPT-text-emd-large | -10.0% | +0.0% | +19.0% | +15.4% | +14.3% | +15.4% | +28.6% | +0.0% | +33.3% | +16.7% | +0.0% | +0.0% | -1.4% | +0.0% | +0.0% | +0.0% | +0.0% | +5.1% |
| | | KaLM-Gemma3-12b-emd | +5.0% | -6.3% | +10.3% | +7.7% | +14.3% | +23.1% | +14.3% | +0.0% | +22.2% | +8.3% | +0.0% | +0.0% | -1.4% | +0.0% | +0.0% | +0.0% | +0.0% | +5.1% |
| | | Qwen3-8b-emd | -10.0% | -4.2% | +23.8% | +23.1% | +14.3% | +23.1% | +14.3% | +6.0% | +22.2% | +8.3% | +0.0% | +0.0% | -1.4% | +0.5% | -0.7% | -0.3% | +0.0% | +6.3% |
| | Alignment | Hint | +11.1% | +11.4% | +72.2% | +56.0% | +0.0% | +4.4% | -46.2% | +0.0% | +5.9% | +4.2% | +9.8% | +11.4% | +0.0% | -1.5% | +0.9% | +0.2% | +1.8% | +2.1% |
| | | GPT-text-emd-large | +0.0% | +9.1% | +61.1% | +36.0% | +4.9% | +2.2% | -7.7% | +2.9% | +52.9% | +29.2% | +9.8% | +11.4% | +20.5% | +5.5% | +1.7% | +0.2% | +3.0% | +3.1% |
| | | KaLM-Gemma3-12b-emd | +22.2% | +0.0% | +61.1% | +40.0% | +34.1% | +28.9% | -23.1% | +8.8% | +35.3% | +20.8% | +17.1% | +22.7% | +13.8% | +3.7% | -0.6% | -2.0% | +2.3% | +1.8% |
| | | Qwen3-8b-emd | +0.0% | +2.3% | +55.6% | +36.0% | +24.4% | +22.2% | +0.0% | +14.7% | +64.7% | +37.5% | +9.8% | +13.6% | +26.7% | +4.8% | +2.4% | +0.0% | +3.2% | +2.3% |
| Qwen3-8b (FullFT) | Zero-shot | Few-shot | -4.8% | +2.0% | +13.9% | +2.4% | -3.7% | -1.8% | +16.7% | +6.8% | -9.1% | -5.0% | -6.0% | -5.3% | -16.5% | -8.3% | -15.4% | -11.7% | -14.3% | -10.0% |
| | | Hint | -14.3% | +2.0% | +5.6% | +0.0% | -5.6% | -5.5% | +11.1% | +4.5% | -3.0% | -5.0% | +0.0% | -1.8% | -13.2% | -6.2% | -9.2% | -8.4% | -7.4% | -4.4% |
| | | GPT-text-emd-large | -14.3% | -2.0% | -13.9% | -14.3% | -1.9% | -1.8% | +5.6% | +9.1% | -9.1% | -10.0% | -8.0% | -3.5% | -11.5% | -6.5% | -11.1% | -10.1% | -12.6% | -11.4% |
| | | KaLM-Gemma3-12b-emd | -4.8% | -5.9% | -11.1% | -16.7% | +3.7% | +3.6% | -22.2% | +6.8% | -15.2% | -7.5% | -2.0% | -1.8% | -8.2% | -6.4% | -14.8% | -10.6% | -9.9% | -7.0% |
| | | Qwen3-8b-emd | -23.8% | -5.9% | -11.1% | -14.3% | +0.0% | -1.8% | -11.1% | +6.8% | +3.0% | +2.5% | -4.0% | -1.8% | -5.8% | -6.7% | -13.0% | -10.1% | -20.5% | -13.2% |
| | Few-shot | Few-shot | +0.0% | -2.0% | +6.4% | +4.8% | +2.5% | -1.8% | +3.7% | +4.5% | +6.1% | +2.5% | +2.4% | +1.8% | +5.1% | +5.4% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | Hint | -2.5% | -2.0% | +0.0% | +0.0% | +0.0% | -1.8% | +3.7% | +4.5% | +6.1% | +2.5% | +0.0% | -1.8% | +0.0% | +5.4% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | GPT-text-emd-large | -2.5% | -2.0% | +0.0% | +0.0% | +2.5% | -1.8% | +3.7% | +9.1% | +6.1% | +2.5% | +2.4% | +1.8% | +5.1% | +10.8% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | KaLM-Gemma3-12b-emd | +0.0% | -2.0% | +0.0% | -4.8% | +2.5% | -1.8% | +0.0% | +4.5% | +0.0% | -2.5% | +2.4% | +1.8% | +0.0% | +5.4% | +0.0% | +0.0% | +0.0% | +0.0% |
| | | Qwen3-8b-emd | -2.5% | -2.0% | +3.2% | +0.0% | +0.0% | -1.8% | +0.0% | +4.5% | +6.1% | +2.5% | +0.0% | -1.8% | +0.0% | +5.4% | +0.0% | +0.0% | +0.0% | +0.0% |

## Box 1: System prompt of itinerary perturbation for DeepSeek V3.2 and R3.2 (ADD operation)

```
You are an expert in Itinerary Perturbation.
Your task must strictly follow these steps for the ADD operation:

1. Definition of an itinerary
An itinerary is a sequence of visiting activities, where each activity is represented as:
[POI name, POI category, longitude, latitude, popularity level].

2. Perturbation operation (ADD)
Insert exactly ONE new POI from the candidate POIs into the original itinerary at any valid position.
This addition must disrupt the itinerary in terms of at least one of the following aspects:
- Spatial distance consistency
- Popularity consistency
- Category diversity consistency

3. Definitions of Consistency and Disruption
(3.1) Category Diversity (CD)
- Let k = number of distinct POI categories (case-insensitive), n = number of POIs.
- Rule:
  If k == 1, then CD = 0.
  Else, CD = k / n.
- For ADD:
  categories_before = categories of the original itinerary
  categories_after = categories_before plus the inserted POI's category
  CD disruption occurs if and only if CD_before != CD_after.

Debug outputs for CD (MUST include):
"categories_raw_before", "categories_set_before",
"categories_raw_after", "categories_set_after",
"cd_before", "cd_after", "cd_disruption".

(3.2) Popularity Consistency
- Popularity levels: {High, Medium, Low}.
- Let P = popularity level list before ADD, Q = after ADD.
- Compute:
  * Hellinger Distance (H) between normalized frequency distributions.
  * Kendall's Tau-b based on category counts -> assign ranks (higher count = smaller rank, ties share rank).
- Tau-b output rule:
  * If computed, output numeric value.
  * If skipped because H > 0.1 already triggered disruption, output "skipped".
- Popularity disruption occurs if H > 0.1 OR Tau-b < 1.

(3.3) Spatial Distance Consistency
- Distances computed between consecutive POIs using the Haversine formula.

Haversine - Precision Rules (MUST follow)
- Use Haversine exactly: R = 6371.0 km; all trig in radians, rad = deg * pi/180.
- Compute dphi = phi2 - phi1 and dlambda = lambda2 - lambda1 in radians at FULL precision. No intermediate rounding or "approximation".
- Half-angle only: a = sin^2(dphi/2) + cos(phi1) * cos(phi2) * sin^2(dlambda/2);
  c = 2 * atan2(sqrt(a), sqrt(1-a)); d = R * c.
  Do NOT drop the leading "2" in c; do NOT replace sin^2(d/2) with d^2. (Only if |d| < 1e-3 may you use (d/2)^2 as a small-angle
  approximation.)
- Round distances only at final display (<= 2 decimals). Segment classification must use the UNROUNDED d.

- Classification:
  Low: < {}m, Medium: {}-{}m, High: > {}m.
- Let P = spatial category list before ADD, Q = after ADD.
- Compute:
  * Hellinger Distance (H) between normalized frequency distributions.
  * Kendall's Tau-b based on category counts -> assign ranks (higher count = smaller rank, ties share rank).
- Tau-b output rule: same as Popularity.
- Spatial disruption occurs if H > 0.1 OR Tau-b < 1.

Debug outputs for Spatial Distance (MUST include):
"spatial_distances_before", "spatial_categories_before",
"spatial_distances_after", "spatial_categories_after",
"spatial_H", "spatial_tau_b", "spatial_disruption".

4. Comparison and Intent Matching
Input includes a "Target Intent Count": 1|2|3.
- Always try to generate a perturbation that yields exactly this number of valid intents.
- If it is possible, output that exact number.
- If it is not possible, output the actual number of valid intents instead.
- If no valid disruption exists, output "UNKNOWN".
```

**Box 1 (continued): System prompt of itinerary perturbation for DeepSeek V3.2 and R3.2 (ADD operation)**

```
INTENT SELECTION RULE (STRICT):
- If cd_disruption = true -> include "Category diversity disruption"
- If popularity_disruption = true -> include "Popularity disruption"
- If spatial_disruption = true -> include "Spatial distance disruption"
- Intents must equal exactly the set of all disruptions flagged true.
- Only output "UNKNOWN" if ALL disruption flags are false.

5. Final Verification
Verify whether the perturbed itinerary satisfies the chosen intent(s).
Validate all numeric values:
cd_before and cd_after must be within [0.0, 1.0].
popularity_H and spatial_H must be within [0.0, 1.0].
popularity_tau_b and spatial_tau_b must be within [-1.0, 1.0] if numeric.
If any value falls outside these ranges:
Recompute CD, Hellinger distances, and Tau-b strictly following the rules in Section 3.
Re-evaluate disruption flags and intents accordingly.
If after recomputation the values are still invalid, replace the entire result with "UNKNOWN".

6. Output format
You MUST return the result in strict JSON format.
{
 "Perturbed Itinerary": [...],
 "Intents": [...],

 -- Category Debug --
 "categories_raw_before": [...],
 "categories_set_before": [...],
 "categories_raw_after": [...],
 "categories_set_after": [...],
 "cd_before": <float>,
 "cd_after": <float>,
 "cd_disruption": true|false,

 -- Popularity Debug --
 "popularity_distribution_before": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_distribution_after": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_ranks_before": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_ranks_after": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_H": <float>,
 "popularity_tau_b": <float or "skipped">,
 "popularity_disruption": true|false,

 -- Spatial Debug --
 "spatial_distances_before": [<float km>, ...],
 "spatial_categories_before": ["Low"|"Medium"|"High", ...],
 "spatial_distances_after": [<float km>, ...],
 "spatial_categories_after": ["Low"|"Medium"|"High", ...],
 "spatial_H": <float>,
 "spatial_tau_b": <float or "skipped">,
 "spatial_disruption": true|false
}
```

## Box 2: System prompt of itinerary perturbation for DeepSeek V3.2 and R3.2 (DELETE operation)

```
You are an expert in Itinerary Perturbation.
Your task must strictly follow these steps for the DELETE operation:

1. Definition of an itinerary
An itinerary is a sequence of visiting activities, where each activity is represented as:
[POI name, POI category, longitude, latitude, popularity level].

2. Perturbation operation (DELETE)
Remove exactly ONE POI from the original itinerary.
This deletion must disrupt the itinerary in terms of at least one of the following aspects:
- Spatial distance consistency
- Popularity consistency
- Category diversity consistency

3. Definitions of Consistency and Disruption
(3.1) Category Diversity (CD)
- Let k = number of distinct POI categories (case-insensitive), n = number of POIs.
- Rule:
  If k == 1, then CD = 0.
  Else, CD = k / n.

- For DELETE:
  categories_before = categories of the original itinerary
  categories_after = categories_before minus the deleted POI's category (if no other POI of that category remains).
  CD disruption occurs if and only if CD_before != CD_after.

Debug outputs for CD (MUST include):
"categories_raw_before", "categories_set_before",
"categories_raw_after", "categories_set_after",
"cd_before", "cd_after", "cd_disruption".

(3.2) Popularity Consistency
- Popularity levels: {High, Medium, Low}.
- Let P = popularity level list before DELETE, Q = after DELETE.
- Compute:
  * Hellinger Distance (H) between normalized frequency distributions.
  * Kendall's Tau-b based on category counts -> assign ranks (higher count = smaller rank, ties share rank).
- Tau-b output rule:
  * If computed, output numeric value.
  * If skipped because H > 0.1 already triggered disruption, output "skipped".
- Popularity disruption occurs if H > 0.1 OR Tau-b < 1.

(3.3) Spatial Distance Consistency
- Distances computed between consecutive POIs using the Haversine formula.
- Classification:
  Low: < {}m, Medium: {}-{}m, High: > {}m.
- Let P = spatial category list before DELETE, Q = after DELETE.
- Compute:
  * Hellinger Distance (H) between normalized frequency distributions.
  * Kendall's Tau-b based on category counts -> assign ranks (higher count = smaller rank, ties share rank).
- Tau-b output rule: same as Popularity.
- Spatial disruption occurs if H > 0.1 OR Tau-b < 1.

Debug outputs for Spatial Distance (MUST include):
"spatial_distances_before", "spatial_categories_before",
"spatial_distances_after", "spatial_categories_after",
"spatial_H", "spatial_tau_b", "spatial_disruption".

4. Comparison and Intent Matching
Compare the original and perturbed itineraries.
From the candidate intents, select only those that are truly supported by the above rules.
If none apply, output "UNKNOWN".

INTENT SELECTION RULE (STRICT):
- If cd_disruption = true -> include "Category diversity disruption"
- If popularity_disruption = true -> include "Popularity disruption"
- If spatial_disruption = true -> include "Spatial distance disruption"
- Intents must equal exactly the set of all disruptions flagged true.
- Only output "UNKNOWN" if ALL disruption flags are false.

5. Final Verification
Verify whether the perturbed itinerary satisfies the chosen intent(s).
Validate all numeric values:
```

**Box 2 (continued): System prompt of itinerary perturbation for DeepSeek V3.2 and R3.2 (DELETE operation)**

```
cd_before and cd_after must be within [0.0, 1.0].
popularity_H and spatial_H must be within [0.0, 1.0].
popularity_tau_b and spatial_tau_b must be within [-1.0, 1.0] if numeric.
If any value falls outside these ranges:
Recompute CD, Hellinger distances, and Tau-b strictly following the rules in Section 3.
Re-evaluate disruption flags and intents accordingly.
If after recomputation the values are still invalid, replace the entire result with "UNKNOWN".

6. Output format
You MUST return the result in strict JSON format.
{
 "Perturbed Itinerary": [...],
 "Intents": [...],

 -- Category Debug --
 "categories_raw_before": [...],
 "categories_set_before": [...],
 "categories_raw_after": [...],
 "categories_set_after": [...],
 "cd_before": <float>,
 "cd_after": <float>,
 "cd_disruption": true|false,

 -- Popularity Debug --
 "popularity_distribution_before": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_distribution_after": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_ranks_before": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_ranks_after": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_H": <float>,
 "popularity_tau_b": <float or "skipped">,
 "popularity_disruption": true|false,

 -- Spatial Debug --
 "spatial_distances_before": [<float km>, ...],
 "spatial_categories_before": ["Low"|"Medium"|"High", ...],
 "spatial_distances_after": [<float km>, ...],
 "spatial_categories_after": ["Low"|"Medium"|"High", ...],
 "spatial_H": <float>,
 "spatial_tau_b": <float or "skipped">,
 "spatial_disruption": true|false
}
```

**Box 3: System prompt of itinerary perturbation for DeepSeek V3.2 and R3.2 (REPLACE operation)**

```
You are an expert in Itinerary Perturbation.
Your task must strictly follow these steps for the REPLACE operation:

1. Definition of an itinerary
An itinerary is a sequence of visiting activities, where each activity is represented as:
[POI name, POI category, longitude, latitude, popularity level].

2. Perturbation operation (REPLACE)
Replace exactly ONE POI in the original itinerary with ONE new POI from the candidate POIs.
This replacement must disrupt the itinerary in terms of at least one of the following aspects:
- Spatial distance consistency
- Popularity consistency
- Category diversity consistency

3. Definitions of Consistency and Disruption
(3.1) Category Diversity (CD)
- Let k = number of distinct POI categories (case-insensitive), n = number of POIs.
- Rule:
  If k == 1, then CD = 0.
  Else, CD = k / n.
- For REPLACE:
  categories_before = categories of the original itinerary
  categories_after = categories_before but with replaced POI's category substituted
- Sets and ratios:
  CD_before = |set(categories_before)| / len(original_itinerary)
  CD_after  = |set(categories_after)| / len(perturbed_itinerary)
  CD disruption occurs if:
    * The CD ratio changes

Debug outputs for CD (MUST include):
"categories_raw_before", "categories_set_before",
"categories_raw_after", "categories_set_after",
"cd_before", "cd_after", "cd_disruption".

(3.2) Popularity Consistency
- Popularity levels: {High, Medium, Low}.
- Let P = popularity level list before REPLACE, Q = after REPLACE.
- Compute:
  * Hellinger Distance (H) between normalized frequency distributions.
  * Kendall's Tau-b based on category counts -> assign ranks (higher count = smaller rank, ties share rank).
- Tau-b output rule:
  * If computed, output numeric value.
  * If skipped because H > 0.1 already triggered disruption, output "skipped".
- Popularity disruption occurs if H > 0.1 OR Tau-b < 1.

(3.3) Spatial Distance Consistency
- Distances computed between consecutive POIs using the Haversine formula.
- Classification:
  Low: < {}m, Medium: {}-{}m, High: > {}m.
- Let P = spatial category list before REPLACE, Q = after REPLACE.
- Compute:
  * Hellinger Distance (H) between normalized frequency distributions.
  * Kendall's Tau-b based on category counts -> assign ranks (higher count = smaller rank, ties share rank).
- Tau-b output rule: same as Popularity.
- Spatial disruption occurs if H > 0.1 OR Tau-b < 1.

Debug outputs for Spatial Distance (MUST include):
"spatial_distances_before", "spatial_categories_before",
"spatial_distances_after", "spatial_categories_after",
"spatial_H", "spatial_tau_b", "spatial_disruption".

4. Comparison and Intent Matching
Input includes a "Target Intent Count": 1|2|3.
- Always try to generate a perturbation that yields exactly this number of valid intents.
- If it is possible, output that exact number.
- If it is not possible, output the actual number of valid intents instead.
- If no valid disruption exists, output "UNKNOWN".

INTENT SELECTION RULE (STRICT):
- If cd_disruption = true -> include "Category diversity disruption"
- If popularity_disruption = true -> include "Popularity disruption"
- If spatial_disruption = true -> include "Spatial distance disruption"
- Intents must equal exactly the set of all disruptions flagged true.
```

**Box 3 (continued): System prompt of itinerary perturbation for DeepSeek V3.2 and R3.2 (REPLACE operation)**

```
- Only output "UNKNOWN" if ALL disruption flags are false.

5. Final Verification
Verify whether the perturbed itinerary satisfies the chosen intent(s).
Validate all numeric values:
cd_before and cd_after must be within [0.0, 1.0].
popularity_H and spatial_H must be within [0.0, 1.0].
popularity_tau_b and spatial_tau_b must be within [-1.0, 1.0] if numeric.
If any value falls outside these ranges:
Recompute CD, Hellinger distances, and Tau-b strictly following the rules in Section 3.
Re-evaluate disruption flags and intents accordingly.
If after recomputation the values are still invalid, replace the entire result with "UNKNOWN".

6. Output format
You MUST return the result in strict JSON format.
{
 "Perturbed Itinerary": [...],
 "Intents": [...],

 -- Category Debug --
 "categories_raw_before": [...],
 "categories_set_before": [...],
 "categories_raw_after": [...],
 "categories_set_after": [...],
 "cd_before": <float>,
 "cd_after": <float>,
 "cd_disruption": true|false,

 -- Popularity Debug --
 "popularity_distribution_before": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_distribution_after": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_ranks_before": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_ranks_after": {"High":<int>, "Medium":<int>, "Low":<int>},
 "popularity_H": <float>,
 "popularity_tau_b": <float or "skipped">,
 "popularity_disruption": true|false,

 -- Spatial Debug --
 "spatial_distances_before": [<float km>, ...],
 "spatial_categories_before": ["Low"|"Medium"|"High", ...],
 "spatial_distances_after": [<float km>, ...],
 "spatial_categories_after": ["Low"|"Medium"|"High", ...],
 "spatial_H": <float>,
 "spatial_tau_b": <float or "skipped">,
 "spatial_disruption": true|false
}
```

## Box 4: System prompt of itinerary perturbation for our V3.2 FM (ADD operation)

```
You are an expert in Itinerary Perturbation.
Your task must strictly follow these steps for the ADD operation.

=== HARD OUTPUT CONTRACT (ABSOLUTE) ===
- SILENT MODE: Output ONLY ONE final JSON object wrapped EXACTLY as:
  <final_json>{ ...STRICT JSON OBJECT... }</final_json>
  * The FIRST non-whitespace characters MUST be "<final_json>"
  * The LAST characters MUST be "</final_json>"
  * NO preamble text, NO trailing text, NO code fences.
- If you cannot complete valid metrics after retries, emit:
  <final_json>{"error":"no_final_json"}</final_json>
- All non-final notes go in <think>...</think> ONLY (brief prose). If tokens are low, SKIP <think>.

0) Tool-Calling Contract (HARD RULES - zero-omission)
- Zero omission: supply ALL required keys exactly; no renaming; no missing args.
- Retry on error: on {"error":...} you MUST re-call the SAME tool with corrected args until success or fallback.
- Case discipline / domains:
  * Popularity labels MUST be Title Case "High","Medium","Low" (domain=["High","Medium","Low"]).
  * Spatial classes MUST be Title Case "Low","Medium","High" (domain=["Low","Medium","High"]).
- ADD length invariant: len(Perturbed Itinerary) == len(Original Itinerary) + 1.
- COPY-THROUGH LOCK (global): Any metric/array/dict returned by a tool is the SINGLE SOURCE OF TRUTH.
  You MUST mirror tool values verbatim into the final JSON fields that bear the same meaning. NEVER fabricate or recalc.
- GEO lock: Call geo_distance_segments EXACTLY ONCE with BOTH waypoint lists. Reuse that first valid result throughout.

0.5) MEMORY (optional but binding if present)
- Input may include {"Memory": ...}. Obey your rules for diversity priority.
- Diversity rule for ADD:
  1) Prefer inserting a POI whose name is NOT in used_poi (exact string match).
  2) Prefer an insertion index i NOT in used_index; if unavoidable, choose among the least-used indices.
  3) If multiple valid insertions meet the Target Intent Count, select one differing from Memory in BOTH POI and index; if still tied,
  sample uniformly at random.
  4) Do NOT change output schema. Any justification stays in <think> only (brief prose).

1) Itinerary Definition
Each activity: [POI name, POI category, longitude, latitude, popularity level].

2) ADD Operation
Insert EXACTLY ONE candidate POI at index i (0..n, with n=len(Original Itinerary)). All other entries remain unchanged and order
preserved.
The insertion must disrupt at least one of:
- Spatial distance consistency
- Popularity consistency
- Category diversity consistency
Build order:
  1) Decide the insertion index i and construct the full Perturbed Itinerary FIRST.
  2) Then call geo_distance_segments EXACTLY ONCE with BOTH lists (use numeric floats for lat/lon). Reuse that FIRST valid result.

3) Consistency Definitions & Echo Contracts

(3.1) Category Diversity (CD) - TOOL-ECHO
- Get categories via TWO calls to categories_from_itinerary:
  cats_before := categories_from_itinerary({"itinerary": Original Itinerary}).categories
  cats_after  := categories_from_itinerary({"itinerary": Perturbed Itinerary}).categories
- HARD SOURCE: "categories_raw_before/after" MUST be EXACT copies of these arrays (token-by-token, including case/punct).
- categories_set_* = unique of categories_raw_* (preserve tokens, order not enforced).
- Echo Gate (must hold BEFORE cd_from_categories and BEFORE <final_json>):
  * len(categories_raw_before) == len(Original Itinerary)
  * len(categories_raw_after)  == len(Perturbed Itinerary)
  * for all i: categories_raw_before[i] == Original Itinerary[i][1]
  * for all i: categories_raw_after[i]  == Perturbed Itinerary[i][1]
- Call cd_from_categories ONLY with those echoed arrays; MIRROR its returns directly into:
  "categories_raw_before","categories_set_before",
  "categories_raw_after","categories_set_after",
  "cd_before","cd_after","cd_disruption".

(3.2) Popularity Consistency - TOOL-ECHO & CANON
- Per-item echo from itinerary col #5 (NO synthesis):
  pop_raw_before := [ Original Itinerary[j][4] for j ]
  pop_raw_after  := [ Perturbed Itinerary[j][4] for j ]
- Canonicalize by mapping {"high":"High","medium":"Medium","low":"Low"} ONLY.
  After canon, EVERY label MUST be in {"High","Medium","Low"}.
- Echo Gate (must hold BEFORE stats_from_categories(popularity)):
```

## Box 4 (continued): System prompt of itinerary perturbation for our V3.2 FM (ADD operation)

```
  * len(pop_raw_before) == len(Original Itinerary)
  * len(pop_raw_after)  == len(Perturbed Itinerary)
  * for all j: pop_raw_before[j] equals Original Itinerary[j][4] up to case normalization ONLY
  * for all j: pop_raw_after[j]  equals Perturbed Itinerary[j][4] up to case normalization ONLY
- Anti-cheat (fatal): labels_* MUST NOT be (a permutation of) the domain UNLESS it exactly matches the itinerary echo.
- Call stats_from_categories with the CANON arrays and domain=["High","Medium","Low"], thresholds={"hellinger":0.1,"tau_b":1.0}.
- MIRROR its returns directly into:
  "popularity_distribution_before","popularity_distribution_after",
  "popularity_ranks_before","popularity_ranks_after",
  "popularity_H","popularity_tau_b","popularity_disruption".

(3.3) Spatial Distance Consistency - TOOL-ECHO & LOCK
- Build waypoints_before/after from itinerary cols (lat,lon) as floats. Call geo_distance_segments ONCE.
- Segment alignment with n=len(Original Itinerary) MUST hold:
  len(spatial_distances_before) == max(0, n-1)
  len(spatial_distances_after)  == max(0, n)
  and len(spatial_categories_*) equals the corresponding distances length.
- Then call stats_from_categories on spatial classes with domain=["Low","Medium","High"], thresholds={"hellinger":0.1,"tau_b":1.0}.
- MIRROR BOTH tool returns directly into ALL spatial debug fields:
  "spatial_distances_before","spatial_categories_before","spatial_ranks_before",
  "spatial_distances_after","spatial_categories_after","spatial_ranks_after",
  "spatial_H","spatial_tau_b","spatial_disruption".

=== MIRROR & ASSERT (HARD KILL-SWITCH) ===
- After all tools succeed and BEFORE emitting <final_json>, you MUST perform these checks:
  1) Equality mirror: For each of the 3 pillars (CD/Popularity/Spatial),
     - The boolean `*_disruption` in your JSON MUST equal the tool field exactly.
     - Each numeric/debug field you output MUST equal the corresponding tool field (same numbers up to normal float formatting).
     If ANY mismatch -> RECONSTRUCT the JSON by directly copying the tool dictionaries into the JSON fields (no free-typing).
     If still mismatched -> emit <final_json>{"error":"no_final_json"}</final_json>.
  2) Ranges: 0<=cd_*<=1, 0<=*_H<=1, -1<=*_tau_b<=1 (numeric). If violated, re-call tool or emit error JSON.
  3) Echo Gates re-check for categories & popularity (lengths and per-item equality to itineraries). If violated, fix and re-call tools.

A) Atomic Intents Builder (MUST OVERWRITE)
- Set:
  cd_flag  := cd_from_categories.cd_disruption
  pop_flag := stats_from_categories(popularity).disruption
  spa_flag := stats_from_categories(spatial).disruption
- Deterministic ordering:
  1) "Category diversity disruption" if cd_flag
  2) "Popularity disruption"        if pop_flag
  3) "Spatial distance disruption"  if spa_flag
- Overwrite JSON["Intents"] with exactly these (or ["UNKNOWN"] if none true). No trimming to match target counts.

4) Target Intent Count (planning only)
- Input "Target Intent Count" is ONLY for choosing index/POI; NEVER appears in output.

5) Final JSON Hygiene
- Only double-quoted strings; no trailing commas; no NaN/Inf; booleans are true/false.
- Keys required (exact names):
{
 "Perturbed Itinerary": [...],
 "Intents": [...],

 "categories_raw_before": [...],
 "categories_set_before": [...],
 "categories_raw_after": [...],
 "categories_set_after": [...],
 "cd_before": <float>,
 "cd_after": <float>,
 "cd_disruption": true|false,

 "popularity_distribution_before": { "High": <float>, "Medium": <float>, "Low": <float> },
 "popularity_distribution_after":  { "High": <float>, "Medium": <float>, "Low": <float> },
 "popularity_ranks_before": { ... },
 "popularity_ranks_after":  { ... },
 "popularity_H": <float>,
 "popularity_tau_b": <float>,
 "popularity_disruption": true|false,

 "spatial_distances_before": [...],
 "spatial_categories_before": [...],
 "spatial_ranks_before": { ... },
```

**Box 4 (continued): System prompt of itinerary perturbation for our V3.2 FM (ADD operation)**

```
 "spatial_distances_after": [...],
 "spatial_categories_after": [...],
 "spatial_ranks_after": { ... },
 "spatial_H": <float>,
 "spatial_tau_b": <float>,
 "spatial_disruption": true|false
}

Output discipline
- Do NOT compute metrics in text; use tools only and MIRROR.
- Immediately before </final_json>, run MIRROR & ASSERT and the Atomic Intents Builder (overwrite JSON["Intents"]).
- If any assertion fails and cannot be corrected by tool re-calls, output <final_json>{"error":"no_final_json"}</final_json>.
```

## Box 5: System prompt of itinerary perturbation for our V3.2 FM (DELETE operation)

```
You are an expert in Itinerary Perturbation.
Your task must strictly follow these steps for the DELETE operation.

=== HARD OUTPUT CONTRACT (ABSOLUTE) ===
- SILENT MODE: Output ONLY ONE final JSON object wrapped EXACTLY as:
  <final_json>{ ...STRICT JSON OBJECT... }</final_json>
  * The FIRST non-whitespace characters MUST be "<final_json>"
  * The LAST characters MUST be "</final_json>"
  * NO preamble text, NO trailing text, NO code fences.
- If you cannot complete valid metrics after retries, emit:
  <final_json>{"error":"no_final_json"}</final_json>
- All non-final notes go in <think>...</think> ONLY (brief prose). If tokens are low, SKIP <think>.

0) Tool-Calling Contract (HARD RULES - zero-omission)
- Zero omission: supply ALL required keys exactly; no renaming; no missing args.
- Retry on error: on {"error":...} you MUST re-call the SAME tool with corrected args until success or fallback.
- Case discipline / domains:
  * Popularity labels MUST be Title Case "High","Medium","Low" (domain=["High","Medium","Low"]).
  * Spatial classes MUST be Title Case "Low","Medium","High" (domain=["Low","Medium","High"]).
- DELETE length invariant: len(Perturbed Itinerary) == len(Original Itinerary) - 1.
- COPY-THROUGH LOCK (global): Any metric/array/dict returned by a tool is the SINGLE SOURCE OF TRUTH.
  You MUST mirror tool values verbatim into the final JSON fields that bear the same meaning. NEVER fabricate or recalc.
- GEO lock: Call geo_distance_segments EXACTLY ONCE with BOTH waypoint lists. Reuse that first valid result throughout.

0.5) MEMORY (optional but binding if present)
- Input may include {"Memory": ...}. Obey your rules for diversity priority.
- Diversity rule for DELETE:
  1) Prefer deleting a POI whose name is NOT in used_poi (exact string match).
  2) Prefer a delete index i NOT in used_index; if unavoidable, choose among the least-used indices.
  3) If multiple valid deletions meet the Target Intent Count, select one differing from Memory in BOTH POI and index; if still tied,
  sample uniformly at random.
  4) Do NOT change output schema. Any justification stays in <think> only (brief prose).

1) Itinerary Definition
Each activity: [POI name, POI category, longitude, latitude, popularity level].

2) DELETE Operation
Delete EXACTLY ONE existing POI at index i (0..n-1, with n=len(Original Itinerary)). All other entries remain unchanged and order
preserved.
The deletion must disrupt at least one of:
- Spatial distance consistency
- Popularity consistency
- Category diversity consistency
Build order:
  1) Decide the deletion index i and construct the full Perturbed Itinerary FIRST.
  2) Then call geo_distance_segments EXACTLY ONCE with BOTH lists (use numeric floats for lat/lon). Reuse that FIRST valid result.

3) Consistency Definitions & Echo Contracts

(3.1) Category Diversity (CD) - TOOL-ECHO
- Get categories via TWO calls to categories_from_itinerary:
  cats_before := categories_from_itinerary({"itinerary": Original Itinerary}).categories
  cats_after  := categories_from_itinerary({"itinerary": Perturbed Itinerary}).categories
- HARD SOURCE: "categories_raw_before/after" MUST be EXACT copies of these arrays (token-by-token, including case/punct).
- categories_set_* = unique of categories_raw_* (preserve tokens, order not enforced).
- Echo Gate (must hold BEFORE cd_from_categories and BEFORE <final_json>):
  * len(categories_raw_before) == len(Original Itinerary)
  * len(categories_raw_after)  == len(Perturbed Itinerary)
  * for all i: categories_raw_before[i] == Original Itinerary[i][1]
  * for all i: categories_raw_after[i]  == Perturbed Itinerary[i][1]
- Call cd_from_categories ONLY with those echoed arrays; MIRROR its returns directly into:
  "categories_raw_before","categories_set_before",
  "categories_raw_after","categories_set_after",
  "cd_before","cd_after","cd_disruption".

(3.2) Popularity Consistency - TOOL-ECHO & CANON
- Per-item echo from itinerary col #5 (NO synthesis):
  pop_raw_before := [ Original Itinerary[j][4] for j ]
  pop_raw_after  := [ Perturbed Itinerary[j][4] for j ]
- Canonicalize by mapping {"high":"High","medium":"Medium","low":"Low"} ONLY.
  After canon, EVERY label MUST be in {"High","Medium","Low"}.
- Echo Gate (must hold BEFORE stats_from_categories(popularity)):
```

**Box 5 (continued): System prompt of itinerary perturbation for our V3.2 FM (DELETE operation)**

```
  * len(pop_raw_before) == len(Original Itinerary)
  * len(pop_raw_after)  == len(Perturbed Itinerary)
  * for all j: pop_raw_before[j] equals Original Itinerary[j][4] up to case normalization ONLY
  * for all j: pop_raw_after[j]  equals Perturbed Itinerary[j][4] up to case normalization ONLY
- Anti-cheat (fatal): labels_* MUST NOT be (a permutation of) the domain UNLESS it exactly matches the itinerary echo.
- Call stats_from_categories with the CANON arrays and domain=["High","Medium","Low"], thresholds={"hellinger":0.1,"tau_b":1.0}.
- MIRROR its returns directly into:
  "popularity_distribution_before","popularity_distribution_after",
  "popularity_ranks_before","popularity_ranks_after",
  "popularity_H","popularity_tau_b","popularity_disruption".

(3.3) Spatial Distance Consistency - TOOL-ECHO & LOCK
- Build waypoints_before/after from itinerary cols (lat,lon) as floats. Call geo_distance_segments ONCE.
- Segment alignment with n=len(Original Itinerary) MUST hold:
  len(spatial_distances_before) == max(0, n-1)
  len(spatial_distances_after)  == max(0, (n-1)-1)   # i.e., max(0, n-2)
  and len(spatial_categories_*) equals the corresponding distances length.
- Then call stats_from_categories on spatial classes with domain=["Low","Medium","High"], thresholds={"hellinger":0.1,"tau_b":1.0}.
- MIRROR BOTH tool returns directly into ALL spatial debug fields:
  "spatial_distances_before","spatial_categories_before","spatial_ranks_before",
  "spatial_distances_after","spatial_categories_after","spatial_ranks_after",
  "spatial_H","spatial_tau_b","spatial_disruption".

=== MIRROR & ASSERT (HARD KILL-SWITCH) ===
- After all tools succeed and BEFORE emitting <final_json>, you MUST perform these checks:
  1) Equality mirror: For each of the 3 pillars (CD/Popularity/Spatial),
     - The boolean `*_disruption` in your JSON MUST equal the tool field exactly.
     - Each numeric/debug field you output MUST equal the corresponding tool field (same numbers up to normal float formatting).
     If ANY mismatch -> RECONSTRUCT the JSON by directly copying the tool dictionaries into the JSON fields (no free-typing).
     If still mismatched -> emit <final_json>{"error":"no_final_json"}</final_json>.
  2) Ranges: 0<=cd_*<=1, 0<=*_H<=1, -1<=*_tau_b<=1 (numeric). If violated, re-call tool or emit error JSON.
  3) Echo Gates re-check for categories & popularity (lengths and per-item equality to itineraries). If violated, fix and re-call tools.

A) Atomic Intents Builder (MUST OVERWRITE)
- Set:
  cd_flag  := cd_from_categories.cd_disruption
  pop_flag := stats_from_categories(popularity).disruption
  spa_flag := stats_from_categories(spatial).disruption
- Deterministic ordering:
  1) "Category diversity disruption" if cd_flag
  2) "Popularity disruption"        if pop_flag
  3) "Spatial distance disruption"  if spa_flag
- Overwrite JSON["Intents"] with exactly these (or ["UNKNOWN"] if none true). No trimming to match target counts.

4) Target Intent Count (planning only)
- Input "Target Intent Count": 1|2|3. It is ONLY a planning constraint to help you choose index/POI; NEVER appear in output.

5) Final JSON Hygiene
- Only double-quoted strings; no trailing commas; no NaN/Inf; booleans are true/false.
- Keys required (exact names):
{
 "Perturbed Itinerary": [...],
 "Intents": [...],

 "categories_raw_before": [...],
 "categories_set_before": [...],
 "categories_raw_after": [...],
 "categories_set_after": [...],
 "cd_before": <float>,
 "cd_after": <float>,
 "cd_disruption": true|false,

 "popularity_distribution_before": { "High": <float>, "Medium": <float>, "Low": <float> },
 "popularity_distribution_after":  { "High": <float>, "Medium": <float>, "Low": <float> },
 "popularity_ranks_before": { ... },
 "popularity_ranks_after":  { ... },
 "popularity_H": <float>,
 "popularity_tau_b": <float>,
 "popularity_disruption": true|false,

 "spatial_distances_before": [...],
 "spatial_categories_before": [...],
 "spatial_ranks_before": { ... },
```

**Box 5 (continued): System prompt of itinerary perturbation for our V3.2 FM (DELETE operation)**

```
 "spatial_distances_after": [...],
 "spatial_categories_after": [...],
 "spatial_ranks_after": { ... },
 "spatial_H": <float>,
 "spatial_tau_b": <float>,
 "spatial_disruption": true|false
}

Output discipline
- Do NOT compute metrics in text; use tools only and MIRROR.
- Immediately before </final_json>, run MIRROR & ASSERT and the Atomic Intents Builder (overwrite JSON["Intents"]).
- If any assertion fails and cannot be corrected by tool re-calls, output <final_json>{"error":"no_final_json"}</final_json>.
```

## Box 6: System prompt of itinerary perturbation for our V3.2 FM (REPLACE operation)

```
You are an expert in Itinerary Perturbation.
Your task must strictly follow these steps for the REPLACE operation.

=== HARD OUTPUT CONTRACT (ABSOLUTE) ===
- SILENT MODE: Output ONLY ONE final JSON object wrapped EXACTLY as:
  <final_json>{ ...STRICT JSON OBJECT... }</final_json>
  * The FIRST non-whitespace characters MUST be "<final_json>"
  * The LAST characters MUST be "</final_json>"
  * NO preamble text, NO trailing text, NO code fences.
- If you cannot complete valid metrics after retries, emit:
  <final_json>{"error":"no_final_json"}</final_json>
- All non-final notes go in <think>...</think> ONLY (brief prose). If tokens are low, SKIP <think>.

0) Tool-Calling Contract (HARD RULES - zero-omission)
- Zero omission: supply ALL required keys exactly; no renaming; no missing args.
- Retry on error: on {"error":...} you MUST re-call the SAME tool with corrected args until success or fallback.
- Case discipline / domains:
  * Popularity labels MUST be Title Case "High","Medium","Low" (domain=["High","Medium","Low"]).
  * Spatial classes MUST be Title Case "Low","Medium","High" (domain=["Low","Medium","High"]).
- REPLACE length invariant: len(Perturbed Itinerary) == len(Original Itinerary).
- COPY-THROUGH LOCK (global): Any metric/array/dict returned by a tool is the SINGLE SOURCE OF TRUTH.
  You MUST mirror tool values verbatim into the final JSON fields that bear the same meaning. NEVER fabricate or recalc.
- Call order & locking (REPLACE):
  1) Decide the exact index i and construct the full Perturbed Itinerary FIRST (replace exactly one POI at index i with one candidate; all
  other entries unchanged in name/category/coordinates/popularity and order).
  2) Then call geo_distance_segments EXACTLY ONCE with BOTH lists (use numeric floats for lat/lon). Reuse that FIRST valid result; do NOT
  call again or overwrite distances/classes.

0.5) MEMORY (optional but binding if present)
- Input may include {"Memory": ...}. Obey your rules for diversity priority.
- Diversity rule for REPLACE:
  1) Prefer choosing a REPLACE index i NOT in used_index; if unavoidable, choose among the least-used indices.
  2) Prefer selecting a NEW candidate POI whose name is NOT in used_poi (exact string match); if unavoidable, choose among the least-used
  names.
  3) If multiple valid replacements meet the Target Intent Count, select one differing from Memory in BOTH the target index and the new
  POI; if still tied, sample uniformly at random.
  4) Do NOT change output schema. Any justification stays in <think> only (brief prose).

1) Itinerary Definition
Each activity: [POI name, POI category, longitude, latitude, popularity level].

2) REPLACE Operation
Replace EXACTLY ONE existing POI with EXACTLY ONE candidate at the SAME index i. Disrupt at least one of:
- Spatial distance consistency
- Popularity consistency
- Category diversity consistency

3) Consistency Definitions & Echo Contracts

(3.1) Category Diversity (CD) - TOOL-ECHO
- Get categories via TWO calls to categories_from_itinerary:
  cats_before := categories_from_itinerary({"itinerary": Original Itinerary}).categories
  cats_after  := categories_from_itinerary({"itinerary": Perturbed Itinerary}).categories
- HARD SOURCE: "categories_raw_before/after" MUST be EXACT copies of these arrays (token-by-token, including case/punct).
- categories_set_* = unique of categories_raw_* (preserve tokens, order not enforced).
- Echo Gate (must hold BEFORE cd_from_categories and BEFORE <final_json>):
  * len(categories_raw_before) == len(Original Itinerary)
  * len(categories_raw_after)  == len(Perturbed Itinerary)
  * for all i: categories_raw_before[i] == Original Itinerary[i][1]
  * for all i: categories_raw_after[i]  == Perturbed Itinerary[i][1]
- Call cd_from_categories ONLY with those echoed arrays; MIRROR its returns directly into:
  "categories_raw_before","categories_set_before",
  "categories_raw_after","categories_set_after",
  "cd_before","cd_after","cd_disruption".

(3.2) Popularity Consistency - TOOL-ECHO & CANON
- Per-item echo from itinerary col #5 (NO synthesis):
  pop_raw_before := [ Original Itinerary[j][4] for j ]
  pop_raw_after  := [ Perturbed Itinerary[j][4] for j ]
- Canonicalize by mapping {"high":"High","medium":"Medium","low":"Low"} ONLY.
  After canon, EVERY label MUST be in {"High","Medium","Low"}.
- Echo Gate (must hold BEFORE stats_from_categories(popularity)):
```

## Box 6 (continued): System prompt of itinerary perturbation for our V3.2 FM (REPLACE operation)

```
  * len(pop_raw_before) == len(Original Itinerary)
  * len(pop_raw_after)  == len(Perturbed Itinerary)
  * for all j: pop_raw_before[j] equals Original Itinerary[j][4] up to case normalization ONLY
  * for all j: pop_raw_after[j]  equals Perturbed Itinerary[j][4] up to case normalization ONLY
- Anti-cheat (fatal): labels_* MUST NOT be (a permutation of) the domain UNLESS it exactly matches the itinerary echo.
- Call stats_from_categories with the CANON arrays and domain=["High","Medium","Low"], thresholds={"hellinger":0.1,"tau_b":1.0}.
- MIRROR its returns directly into:
  "popularity_distribution_before","popularity_distribution_after",
  "popularity_ranks_before","popularity_ranks_after",
  "popularity_H","popularity_tau_b","popularity_disruption".

(3.3) Spatial Distance Consistency - TOOL-ECHO & LOCK
- Build waypoints_before/after from itinerary cols (lat,lon) as floats. Call geo_distance_segments ONCE.
- Segment alignment with n=len(Original Itinerary) MUST hold:
  len(spatial_distances_before) == len(spatial_distances_after) == max(0, n-1),
  and len(spatial_categories_*) equals the corresponding distances length.
- Then call stats_from_categories on spatial classes with domain=["Low","Medium","High"], thresholds={"hellinger":0.1,"tau_b":1.0}.
- SPATIAL MIRROR CONTRACT (REPLACE):
  * "spatial_distances_before" := geo_result.distances_before
  * "spatial_categories_before" := geo_result.classes_before
  * "spatial_ranks_before" := spatial_stats.ranks_before
  * "spatial_distances_after"  := geo_result.distances_after
  * "spatial_categories_after" := geo_result.classes_after
  * "spatial_ranks_after" := spatial_stats.ranks_after
  * "spatial_H" := spatial_stats.hellinger
  * "spatial_tau_b" := spatial_stats.tau_b
  * "spatial_disruption" := spatial_stats.disruption
  (You MAY round numbers to <=12 decimals; you MUST NOT round a positive Hellinger to 0.0.)

=== MIRROR & ASSERT (HARD KILL-SWITCH) ===
- After all tools succeed and BEFORE emitting <final_json>, you MUST perform these checks:
  1) Equality mirror: For CD/Popularity/Spatial,
     - JSON `*_disruption` MUST equal the tool boolean exactly.
     - Every numeric/debug field MUST equal the tool field (same values up to normal float formatting).
     If ANY mismatch -> RECONSTRUCT the JSON by directly copying the tool dicts (no free-typing).
     If still mismatched -> emit <final_json>{"error":"no_final_json"}</final_json>.
  2) Ranges: 0<=cd_*<=1, 0<=*_H<=1, -1<=*_tau_b<=1 (numeric). If violated, re-call tool or emit error JSON.
  3) Echo Gates re-check for categories & popularity (lengths and per-item equality to itineraries). If violated, fix and re-call tools.
- SPATIAL EQUALITY CHECKLIST (REPLACE):
  Assert deep equality (after allowed rounding) between your JSON spatial fields and the two tool outputs (geo_distance_segments +
  stats_from_categories). If "spatial_categories_before != spatial_categories_after" while tool.hellinger>0 still appears as 0.0 in your
  JSON, discard and rebuild; if unresolved, emit error JSON.

A) Atomic Intents Builder (MUST OVERWRITE)
- Set:
  cd_flag  := cd_from_categories.cd_disruption
  pop_flag := stats_from_categories(popularity).disruption
  spa_flag := stats_from_categories(spatial).disruption
- Deterministic ordering:
  1) "Category diversity disruption" if cd_flag
  2) "Popularity disruption"        if pop_flag
  3) "Spatial distance disruption"  if spa_flag
- Overwrite JSON["Intents"] with exactly these (or ["UNKNOWN"] if none true). No trimming to match target counts.

4) Target Intent Count (planning only)
- Input "Target Intent Count" is ONLY for choosing index/POI; NEVER appears in output.

5) Final JSON Hygiene
- Only double-quoted strings; no trailing commas; no NaN/Inf; booleans are true/false.
- Keys required (exact names):
{
 "Perturbed Itinerary": [...],
 "Intents": [...],

 "categories_raw_before": [...],
 "categories_set_before": [...],
 "categories_raw_after": [...],
 "categories_set_after": [...],
 "cd_before": <float>,
 "cd_after": <float>,
 "cd_disruption": true|false,

 "popularity_distribution_before": { "High": <float>, "Medium": <float>, "Low": <float> },
```

**Box 6 (continued): System prompt of itinerary perturbation for our V3.2 FM (REPLACE operation)**

```
 "popularity_distribution_after":  { "High": <float>, "Medium": <float>, "Low": <float> },
 "popularity_ranks_before": { ... },
 "popularity_ranks_after":  { ... },
 "popularity_H": <float>,
 "popularity_tau_b": <float>,
 "popularity_disruption": true|false,

 "spatial_distances_before": [...],
 "spatial_categories_before": [...],
 "spatial_ranks_before": { ... },
 "spatial_distances_after": [...],
 "spatial_categories_after": [...],
 "spatial_ranks_after": { ... },
 "spatial_H": <float>,
 "spatial_tau_b": <float>,
 "spatial_disruption": true|false
}

Output discipline
- Do NOT compute metrics in text; use tools only and MIRROR.
- Immediately before </final_json>, run MIRROR & ASSERT and the Atomic Intents Builder (overwrite JSON["Intents"]).
- If any assertion fails and cannot be corrected by tool re-calls, output <final_json>{"error":"no_final_json"}</final_json>.
```

**Box 7: Template of memory module**

```
In round {}, the operation is {}.
The POIs you have chosen for {} include {}.
The positions (indexes) you has chosen are {}.
```

## Box 8: System prompt of itinerary modification (ADD operation)

```
You are an itinerary editor with strong reasoning over distributions and ranking patterns.

[Inputs]
You will receive:
1) Need-to-Modify Itinerary (length m): a JSON array of activities, each
[POI name, POI category, longitude, latitude, popularity("High"|"Medium"|"Low")].
2) Candidate POIs: an array of objects with keys "cand_id" and "poi" (where "poi" is the same 5-tuple).
3) Hints: high-level guidance; may mention any subset of axes: Popularity / Categories / Spatial.
4) Spatial thresholds (kilometers): threshold_low_km, threshold_high_km.
Class rule for consecutive-leg distances (Haversine, Earth radius 6371.0088 km):
Low if d < threshold_low_km; High if d > threshold_high_km; Medium otherwise.
[End of Inputs]

[Task]
Perform exactly ONE edit: INSERT one candidate POI at index i in [0..m] to REPAIR the itinerary.
Your edit must satisfy both:
A) On every hinted axis => a distribution shift is present after INSERT vs before INSERT:
- For popularity and spatial axes:
* Mixture shift: Hellinger distance H between normalized frequency distributions > 0.1; OR
* Ranking-order shift: majority=>minority order of counts changes (ties allowed; check internally).
- For category axis (category diversity, CD):
* Define CD = number of unique categories / length of itinerary, except when number of unique categories = 1, set CD = 0.
  If CD_before != CD_after, then category diversity is considered to have changed (a shift is detected).
B) On every non-hinted axis => distribution remains invariant:
- For popularity and spatial axes:
* Mixture invariance: H <= 0.1; AND
* Ranking-order invariance: order unchanged.
- For category axis (category diversity, CD):
* CD_before must equal CD_after (CD_before = CD_after), i.e., category diversity does not change.
[End of Task]

[Guidance]
1) When determining which candidate POI to insert and at which index, first prioritize changes in popularity and category diversity
   (or lack thereof) according to the shift/invariance requirements. Only after evaluating popularity and category diversity
   should you consider changes or invariance in spatial distance. In other words, always give priority to popularity and
   category axes when optimizing for the required shift/invariance before taking spatial distance into account.
2) To reduce process bias, systematically evaluate all combinations of candidates and insertion indices without favoring any
   particular candidate or index order. Avoid always starting reasoning from the first candidate or from the first index.
3) Pay careful attention to selecting the correct insertion index (insert_index) and to evaluating all candidate POIs systematically.
   For robust reasoning, thoroughly test all (candidate, index) pairs for compliance with shift/invariance requirements across hinted
   and non-hinted axes. To avoid bias, do not process candidates or indices in a strictly sequential or default order - ensure every
   candidate and index position is considered equally. Many LLMs exhibit position bias by favoring the first candidate or the first
   index; apply extra rigor to prevent this in both axes of selection.
4) Ensure implementation is robust to edge cases, such as duplicate POIs, itineraries with only one or two elements, or evenly
   distributed axes. Double-check the selected insert_index to confirm that the chosen edit maximizes hinted-axis shift while
   maintaining invariance elsewhere, as required.
[End of Guidance]

[Steps]
1) For all candidate POIs and all possible insert indices i in [0..m], exhaustively simulate each insertion and recompute the itinerary
state:
a) Analyze: parse itinerary, candidates, and Hints; mark hinted vs non-hinted axes.
b) Establish BEFORE state:
- popularity counts over {"High","Medium","Low"};
- category counts using case-insensitive labels for counting {preserve original strings later};
- spatial class sequence over {"Low","Medium","High"} via thresholds; record ranking orders.
- category diversity (CD) = number of unique categories / itinerary length, except when number of unique categories = 1, set CD = 0.
c) Compute the AFTER state after insertion and update CD as well.
2) Search: retain all options that meet A & B.
3) Select (deterministic):
- Prefer options that maximize hinted-axis shift (larger Hellinger distance and/or clear ranking change or CD change) while keeping
  non-hinted deviation minimal (H close to 0, order unchanged, no CD change if not hinted).
- When selecting among options, assess and prioritize changes in popularity and category axes (including CD) before considering changes
  in spatial distance.
- Evaluate all possible (candidate, index) pairs impartially to avoid any sequence or position bias.
4) Output: return exactly ONE JSON object in the specified format; do NOT include analysis, calculations, metric names, or thresholds.
[End of Steps]

[Output]
Return ONLY one JSON object with no code fences, no extra text.
{
"insert_index": <int in [0..m]>,
"selected_cand_id": <string|number>,
```

**Box 8 (continued): System prompt of itinerary modification (ADD operation)**

```
"selected_poi": [name, category, lon, lat, popularity],
"reason": "<Hint-grounded explanation showing shift on hinted axes and invariance on non-hinted axes>",
"confidence": <float 0..1>
}
[End of Output]

[Constraints & Self-check]
- Insert exactly one candidate POI; final length must be m+1.
- Selection must be fully deterministic given the same input; randomization is not permitted.
- Verify:
  - The selected_cand_id must come from the provided candidates.
  - The selected_poi must be an exact, verbatim copy of the candidate's 5-tuple.
- Thoroughly validate hint compliance after the change: for each hinted axis, a required shift (mixture or ranking-order for
popularity/spatial,
  or category diversity change for category) must be present; for each non-hinted axis, check both mixture invariance (Hellinger distance
  <= 0.1)
  and ranking-order invariance, and for category, category diversity remains unchanged.
- Derive spatial classes strictly from inter-POI distances using the provided thresholds (Low/Medium/High).
- When counting categories for diversity or distribution, treat labels case-insensitively for counting but preserve their original
spelling in final output.
- Do not include formulas, thresholds, or intermediate calculation details in the output JSON.
- If no solution fully satisfies every constraint, select the option that achieves the strongest possible shift on hinted axes and
minimizes deviations on
  non-hinted axes; clearly indicate this limitation in the "reason" field.
- Assign "confidence" as follows: 1.0 for a unique, clear solution; 0.7-0.9 for strong but somewhat ambiguous solutions; use lower values
for partial/no
  satisfaction due to unavoidable constraints.
[End of Constraints & Self-check]

[Examples]
(Optional, for guidance only; do NOT copy into the final answer.)
Paste one or more raw JSON example objects here (not an array). If none, leave this section empty.
[End of Examples]
```

## Box 9: System prompt of itinerary modification (DELETE operation)

```
You are an itinerary editor with strong reasoning over distributions and ranking patterns.
[Inputs]
You will receive:
1) Need-to-Modify Itinerary (length m): a JSON array of activities, each
[POI name, POI category, longitude, latitude, popularity("High"|"Medium"|"Low")].
2) Hints: high-level guidance; may mention any subset of axes: Popularity / Categories / Spatial.
3) Spatial thresholds (kilometers): threshold_low_km, threshold_high_km.
Class rule for consecutive-leg distances (Haversine, Earth radius 6371.0088 km):
Low if d < threshold_low_km; High if d > threshold_high_km; Medium otherwise.
[End of Inputs]

[Task]
Perform exactly ONE edit: DELETE one itinerary POI at index i in [0..m-1] to REPAIR the itinerary.
Your edit must satisfy both:
A) On every hinted axis => a distribution shift is present after DELETE vs before DELETE:
- For popularity and spatial axes:
* Mixture shift: Hellinger distance H between normalized frequency distributions > 0.1; OR
* Ranking-order shift: majority=>minority order of counts changes (ties allowed; check internally).
- For category axis (category diversity, CD):
* Define CD = number of unique categories / length of itinerary, except when number of unique categories = 1, set CD = 0.
  If CD_before != CD_after, then category diversity is considered to have changed (a shift is detected).
B) On every non-hinted axis => distribution remains invariant:
- For popularity and spatial axes:
* Mixture invariance: H <= 0.1; AND
* Ranking-order invariance: order unchanged.
- For category axis (category diversity, CD):
* CD_before must equal CD_after (CD_before = CD_after), i.e., category diversity does not change.
[End of Task]

[Guidance]
1) When determining which POI to delete (removed_index), first prioritize changes in popularity and category diversity (or lack thereof)
   according to the shift/invariance requirements. Only after evaluating popularity and category diversity should you consider changes
   or invariance in spatial distance. In other words, always give priority to popularity and category axes when optimizing for the
   required shift/invariance before taking spatial distance into account.
2) To reduce process bias, systematically evaluate all possible removal indices without favoring any particular index order. Avoid always
   starting reasoning from the first index. Ensure every index position is considered equally. Many LLMs exhibit position bias by favoring
   the first index; apply extra rigor to prevent this in your selection.
3) Pay careful attention to selecting the correct removed_index and to evaluating all options systematically. For robust reasoning,
   thoroughly test all index positions for compliance with shift/invariance requirements across hinted and non-hinted axes. To avoid bias,
   do not process indices in a strictly sequential or default order - ensure every index position is considered equally.
4) Ensure implementation is robust to edge cases, such as duplicate POIs, itineraries with only one or two elements, or evenly distributed
   axes. Double-check the selected removed_index to confirm that the chosen edit maximizes hinted-axis shift while maintaining invariance
   elsewhere, as required.
[End of Guidance]

[Steps]
1) For all possible removal indices i in [0..m-1], exhaustively simulate each POI removal and recompute the itinerary state:
a) Analyze: parse itinerary and Hints; mark hinted vs non-hinted axes.
b) Establish BEFORE state:
- popularity counts over {"High","Medium","Low"};
- category counts using case-insensitive labels for counting {preserve original strings later};
- spatial class sequence over {"Low","Medium","High"} via thresholds; record ranking orders.
- category diversity (CD) = number of unique categories / itinerary length, except when number of unique categories = 1, set CD = 0.
c) Compute the AFTER state after removal and update CD as well.
2) Search: retain all options that meet A & B.
3) Select (deterministic):
- Prefer options that maximize hinted-axis shift (larger Hellinger distance and/or clear ranking change or CD change) while keeping
  non-hinted deviation minimal (H close to 0, order unchanged, no CD change if not hinted).
- When selecting among options, assess and prioritize changes in popularity and category axes (including CD) before considering changes
  in spatial distance.
- Evaluate all possible index positions impartially to avoid any sequence or position bias.
4) Output: return exactly ONE JSON object in the specified format; do NOT include analysis, calculations, metric names, or thresholds.
[End of Steps]

[Output]
Return ONLY one JSON object with no code fences, no extra text.
{
"removed_index": <int in [0..m-1]>,
"reason": "<Hint-grounded explanation showing shift on hinted axes and invariance on non-hinted axes>",
"confidence": <float 0..1>
}
[End of Output]
```

**Box 9 (continued): System prompt of itinerary modification (DELETE operation)**

```
[Constraints & Self-check]
- Delete exactly one itinerary POI; length must become m-1.
- Selection must be fully deterministic given the same input; randomization is not permitted.
- Verify:
- The deleted POI index must be from the provided itinerary.
- Thoroughly validate hint compliance after the change: for each hinted axis, a required shift (mixture or ranking-order for
popularity/spatial,
  or category diversity change for category) must be present; for each non-hinted axis, check both mixture invariance (Hellinger distance
  <= 0.1)
  and ranking-order invariance, and for category, category diversity remains unchanged.
- Derive spatial classes strictly from inter-POI distances using the provided thresholds (Low/Medium/High).
- When counting categories for diversity or distribution, treat labels case-insensitively for counting but preserve their original
spelling in final output.
- Do not include formulas, thresholds, or intermediate calculation details in the output JSON.
- If no solution fully satisfies every constraint, select the option that achieves the strongest possible shift on hinted axes and
minimizes deviations on
  non-hinted axes; clearly indicate this limitation in the "reason" field.
- Assign "confidence" as follows: 1.0 for a unique, clear solution; 0.7-0.9 for strong but somewhat ambiguous solutions; use lower values
for partial/no
  satisfaction due to unavoidable constraints.
[End of Constraints & Self-check]

[Examples]
(Optional, for guidance only; do NOT copy into the final answer.)
Paste one or more raw JSON example objects here (not an array). If none, leave this section empty.
[End of Examples]
```

## Box 10: System prompt of itinerary modification (REPLACE operation)

```
You are an itinerary editor with strong reasoning over distributions and ranking patterns.
[Inputs]
You will receive:
1) Need-to-Modify Itinerary (length m): a JSON array of activities, each
[POI name, POI category, longitude, latitude, popularity("High"|"Medium"|"Low")].
2) Candidate POIs: an array of objects with keys "cand_id" and "poi" (where "poi" is the same 5-tuple).
3) Hints: high-level guidance; may mention any subset of axes: Popularity / Categories / Spatial.
4) Spatial thresholds (kilometers): threshold_low_km, threshold_high_km.
Class rule for consecutive-leg distances (Haversine, Earth radius 6371.0088 km):
Low if d < threshold_low_km; High if d > threshold_high_km; Medium otherwise.
[End of Inputs]

[Task]
Perform exactly ONE edit: REPLACE one itinerary POI at index i in [0..m-1] with one candidate POI to REPAIR the itinerary.
Your edit must satisfy both:
A) On every hinted axis => a distribution shift is present after REPLACE vs before REPLACE:
- For popularity and spatial axes:
* Mixture shift: Hellinger distance H between normalized frequency distributions > 0.1; OR
* Ranking-order shift: majority=>minority order of counts changes (ties allowed; check internally).
- For category axis (category diversity, CD):
* Define CD = number of unique categories / length of itinerary, except when number of unique categories = 1, set CD = 0.
  If CD_before != CD_after, then category diversity is considered to have changed (a shift is detected).
B) On every non-hinted axis => distribution remains invariant:
- For popularity and spatial axes:
* Mixture invariance: H <= 0.1; AND
* Ranking-order invariance: order unchanged.
- For category axis (category diversity, CD):
* CD_before must equal CD_after (CD_before = CD_after), i.e., category diversity does not change.
[End of Task]

[Guidance]
1) When determining which candidate POI to use for replacement and at which index, first prioritize changes in popularity and category
diversity
   (or lack thereof) according to the shift/invariance requirements. Only after evaluating popularity and category diversity should you
   consider
   changes or invariance in spatial distance. In other words, always give priority to popularity and category axes when optimizing for
   the required
   shift/invariance before taking spatial distance into account.
2) To reduce process bias, systematically evaluate all combinations of candidates and replacement indices without favoring any particular
candidate
   or index order. Avoid always starting reasoning from the first candidate or from the first index.
3) Pay careful attention to selecting the correct replacement index (replace_index) and to evaluating all candidate POIs systematically.
For robust
   reasoning, thoroughly test all (candidate, index) pairs for compliance with shift/invariance requirements across hinted and non-hinted
   axes. To avoid
   bias, do not process candidates or indices in a strictly sequential or default order - ensure every candidate and index position is
   considered equally.
   Many LLMs exhibit position bias by favoring the first candidate or the first index; apply extra rigor to prevent this in both axes of
   selection.
4) Ensure implementation is robust to edge cases, such as duplicate POIs, itineraries with only one or two elements, or evenly
distributed axes.
   Double-check the selected replace_index to confirm that the chosen edit maximizes hinted-axis shift while maintaining invariance
   elsewhere, as required.
[End of Guidance]

[Steps]
1) For all candidate POIs and all possible replacement indices i in [0..m-1], exhaustively simulate each replacement and recompute the
itinerary state:
a) Analyze: parse itinerary, candidates, and Hints; mark hinted vs non-hinted axes.
b) Establish BEFORE state:
- popularity counts over {"High","Medium","Low"};
- category counts using case-insensitive labels for counting {preserve original strings later};
- spatial class sequence over {"Low","Medium","High"} via thresholds; record ranking orders.
- category diversity (CD) = number of unique categories / itinerary length, except when number of unique categories = 1, set CD = 0.
c) Compute the AFTER state after replacement and update CD as well.
2) Search: retain all options that meet A & B.
3) Select (deterministic):
- Prefer options that maximize hinted-axis shift (larger Hellinger distance and/or clear ranking change or CD change) while keeping
non-hinted deviation
  minimal (H close to 0, order unchanged, no CD change if not hinted).
```

**Box 10 (continued): System prompt of itinerary modification (REPLACE operation)**

```
- When selecting among options, assess and prioritize changes in popularity and category axes (including CD) before considering changes
in spatial distance.
- Evaluate all possible (candidate, index) pairs impartially to avoid any sequence or position bias.
4) Output: return exactly ONE JSON object in the specified format; do NOT include analysis, calculations, metric names, or thresholds.
[End of Steps]

[Output]
Return ONLY one JSON object with no code fences, no extra text.
{
"replace_index": <int in [0..m-1]>,
"selected_cand_id": <string|number>,
"selected_poi": [name, category, lon, lat, popularity],
"reason": "<Hint-grounded explanation showing shift on hinted axes and invariance on non-hinted axes>",
"confidence": <float 0..1>
}
[End of Output]

[Constraints & Self-check]
- Replace exactly one itinerary POI; length must remain m.
- Selection must be fully deterministic given the same input; randomization is not permitted.
- Verify:
  - The selected_cand_id must come from the provided candidates.
  - The selected_poi must be an exact, verbatim copy of the candidate's 5-tuple.
- Thoroughly validate hint compliance after the change: for each hinted axis, a required shift (mixture or ranking-order for
popularity/spatial,
  or category diversity change for category) must be present; for each non-hinted axis, check both mixture invariance (Hellinger distance
  <= 0.1)
  and ranking-order invariance, and for category, category diversity remains unchanged.
- Derive spatial classes strictly from inter-POI distances using the provided thresholds (Low/Medium/High).
- When counting categories for diversity or distribution, treat labels case-insensitively for counting but preserve their original
spelling in final output.
- Do not include formulas, thresholds, or intermediate calculation details in the output JSON.
- If no solution fully satisfies every constraint, select the option that achieves the strongest possible shift on hinted axes and
minimizes deviations on
  non-hinted axes; clearly indicate this limitation in the "reason" field.
- Assign "confidence" as follows: 1.0 for a unique, clear solution; 0.7-0.9 for strong but somewhat ambiguous solutions; use lower values
for partial/no
  satisfaction due to unavoidable constraints.
[End of Constraints & Self-check]

[Examples]
(Optional, for guidance only; do NOT copy into the final answer.)
Paste one or more raw JSON example objects here (not an array). If none, leave this section empty.
[End of Examples]
```

## 1.6 Proof of Lower Bound of Hellinger Distance

As mentioned before, we state that Hellinger distance is sensitive to extreme cases regarding the macro view. For instance, given an itinerary with length $n$ that includes exactly one low-popularity POI, $a$ high-popularity POIs and $n-a-1$ medium-popularity POIs, an effective perturbation is to delete the low-popularity POI, which aligns with the intent $z_{pop}$. In section 4.2, we set the threshold of Hellinger distance for disruption identification (*i.e.,* $\theta$) as 0.1. Here, we aim to theoretically verify this value can identify all the aforementioned extreme cases. In this case, we can define the popularity distribution of original itinerary and perturbed itinerary as $P = \left(\frac{a}{n}, \frac{n-a-1}{n}, \frac{1}{n}\right), Q = \left(\frac{a}{n-1}, \frac{n-a-1}{n-1}, 0\right)$, respectively. Based on it, we have the following lemma to obtain the lower bound of Hellinger distance for measuring these two distributions.

Lemma 1.1 (Lower bound of the Hellinger distance under the DELETE operation). *Let*

$$P = \left(\frac{a}{n}, \frac{n-a-1}{n}, \frac{1}{n}\right), \qquad Q = \left(\frac{a}{n-1}, \frac{n-a-1}{n-1}, 0\right),$$

*where $n \geq 2$ and $0 \leq a \leq n-1$. Then the squared Hellinger distance satisfies*

$$H^2(P,Q) = 1 - \sqrt{\frac{n-1}{n}} \quad \text{and hence} \quad H(P,Q) \geq \frac{1}{\sqrt{2n}}.$$

Proof. Recall the squared Hellinger distance

$$H^2(P,Q) = 1 - \sum_{i=1}^{3} \sqrt{p_i q_i}.$$

We compute the Bhattacharyya coefficient:

$$\sum_{i=1}^{3} \sqrt{p_i q_i} = \sqrt{\frac{a}{n} \cdot \frac{a}{n-1}} + \sqrt{\frac{n-a-1}{n} \cdot \frac{n-a-1}{n-1}} + \sqrt{\frac{1}{n} \cdot 0}.$$

The last term is zero. For the first two terms,

$$\sqrt{\frac{a}{n} \cdot \frac{a}{n-1}} = \frac{a}{\sqrt{n(n-1)}}, \qquad \sqrt{\frac{n-a-1}{n} \cdot \frac{n-a-1}{n-1}} = \frac{n-a-1}{\sqrt{n(n-1)}}.$$

Therefore,

$$\sum_{i=1}^{3} \sqrt{p_i q_i} = \frac{a + (n-a-1)}{\sqrt{n(n-1)}} = \frac{n-1}{\sqrt{n(n-1)}} = \sqrt{\frac{n-1}{n}},$$

and substituting back yields

$$H^2(P,Q) = 1 - \sqrt{\frac{n-1}{n}} = 1 - \sqrt{1 - \frac{1}{n}}.$$

To derive a lower bound, we use the identity

$$1 - \sqrt{1-x} = \frac{x}{1+\sqrt{1-x}}, \qquad x \in [0,1].$$

Setting $x = \frac{1}{n}$ gives

$$H^2(P,Q) = \frac{\frac{1}{n}}{1 + \sqrt{1 - \frac{1}{n}}} \geq \frac{\frac{1}{n}}{2} = \frac{1}{2n},$$

since $1 + \sqrt{1 - \frac{1}{n}} \leq 2$. Taking square roots completes the proof:

$$H(P,Q) \geq \sqrt{\frac{1}{2n}} = \frac{1}{\sqrt{2n}}.$$

□

Based on Lemma A.1, we can find that with the increase of itinerary length $n$, the lower bound of Hellinger distance decreases. To guarantee that the threshold $\theta$ enables Hellinger distance to identify disruption, we have to satisfy $\theta \leq 1/\sqrt{2n}$, and we have $n \leq 1/2\theta^2$. In section 4.2, we set $\theta$ to 0.1, and we have $n \leq 50$. This means that for any itinerary whose length is no more than 50, Hellinger distance can identify disruption in extreme cases. In our dataset, the maximum of itinerary length is 21, which verifies the effectiveness of setting $\theta$ to 0.1.

It is noted that the aforementioned case is design under the DELETE operation. We further generalize it to the ADD and REPLACE operations.

Lemma 1.2 (Lower bound of the Hellinger distance under the ADD operation). *Let*

$$P = \left(\frac{a}{n}, \frac{n-a}{n}, 0\right), \qquad Q = \left(\frac{a}{n+1}, \frac{n-a}{n+1}, \frac{1}{n+1}\right),$$

*where $n \geq 1$ and $0 \leq a \leq n$. Then the squared Hellinger distance satisfies*

$$H^2(P,Q) = 1 - \sqrt{\frac{n}{n+1}} \quad \text{and hence} \quad H(P,Q) \geq \frac{1}{\sqrt{2(n+1)}}.$$

Proof. Recall the squared Hellinger distance

$$H^2(P,Q) = 1 - \sum_{i=1}^{3} \sqrt{p_i q_i}.$$

We compute the Bhattacharyya coefficient:

$$\sum_{i=1}^{3} \sqrt{p_i q_i} = \sqrt{\frac{a}{n} \cdot \frac{a}{n+1}} + \sqrt{\frac{n-a}{n} \cdot \frac{n-a}{n+1}} + \sqrt{0 \cdot \frac{1}{n+1}}.$$

The last term is zero. For the first two terms,

$$\sqrt{\frac{a}{n} \cdot \frac{a}{n+1}} = \frac{a}{\sqrt{n(n+1)}}, \qquad \sqrt{\frac{n-a}{n} \cdot \frac{n-a}{n+1}} = \frac{n-a}{\sqrt{n(n+1)}}.$$

Therefore,

$$\sum_{i=1}^{3} \sqrt{p_i q_i} = \frac{a + (n-a)}{\sqrt{n(n+1)}} = \frac{n}{\sqrt{n(n+1)}} = \sqrt{\frac{n}{n+1}},$$

and substituting back yields

$$H^2(P,Q) = 1 - \sqrt{\frac{n}{n+1}} = 1 - \frac{1}{\sqrt{1+\frac{1}{n}}}.$$

To derive a lower bound, we use the identity

$$1 - \frac{1}{\sqrt{1+x}} = \frac{x}{\sqrt{1+x}\,\left(1+\sqrt{1+x}\right)}, \qquad x \geq 0.$$

Setting $x = \frac{1}{n}$ gives

$$H^2(P,Q) = \frac{\frac{1}{n}}{\sqrt{1+\frac{1}{n}}\,\left(1+\sqrt{1+\frac{1}{n}}\right)} \geq \frac{\frac{1}{n}}{2\left(1+\frac{1}{n}\right)} = \frac{1}{2(n+1)},$$

since $\sqrt{1+\frac{1}{n}} \leq 1 + \frac{1}{n}$ and thus $\sqrt{1+\frac{1}{n}}\left(1+\sqrt{1+\frac{1}{n}}\right) \leq \left(1+\frac{1}{n}\right) \cdot 2\left(1+\frac{1}{n}\right) = 2\left(1+\frac{1}{n}\right)^2$, in particular it is at most $2\left(1+\frac{1}{n}\right)$ for $n \geq 1$. Taking square roots completes the proof:

$$H(P,Q) \geq \sqrt{\frac{1}{2(n+1)}} = \frac{1}{\sqrt{2(n+1)}}.$$

□

When $\theta = 0.1$, we have $n \leq 49$. In our dataset, the maximum of itinerary length is only 21, which verifies the effectiveness of setting $\theta$ to 0.1.

Lemma 1.3 (Lower bound of the Hellinger distance under the REPLACE operation). *Let*

$$P = \left(\frac{a}{n}, \frac{n-a}{n}, 0\right), \qquad Q = \left(\frac{a-1}{n}, \frac{n-a}{n}, \frac{1}{n}\right),$$

*where $n \geq 2$ and $1 \leq a \leq n$. Then the squared Hellinger distance satisfies*

$$H^2(P,Q) = 1 - \frac{\sqrt{a(a-1)} + (n-a)}{n} \quad \textit{and hence} \quad H(P,Q) \geq \frac{1}{\sqrt{2n}}.$$

Proof. Recall the squared Hellinger distance

$$H^2(P,Q) = 1 - \sum_{i=1}^{3} \sqrt{p_i q_i}.$$

We compute the Bhattacharyya coefficient:

$$\sum_{i=1}^{3} \sqrt{p_i q_i} = \sqrt{\frac{a}{n} \cdot \frac{a-1}{n}} + \sqrt{\frac{n-a}{n} \cdot \frac{n-a}{n}} + \sqrt{0 \cdot \frac{1}{n}}.$$

The last term is zero. For the first two terms,

$$\sqrt{\frac{a}{n} \cdot \frac{a-1}{n}} = \frac{\sqrt{a(a-1)}}{n}, \qquad \sqrt{\frac{n-a}{n} \cdot \frac{n-a}{n}} = \frac{n-a}{n}.$$

Therefore,

$$\sum_{i=1}^{3} \sqrt{p_i q_i} = \frac{\sqrt{a(a-1)} + (n-a)}{n},$$

and substituting back yields

$$H^2(P,Q) = 1 - \frac{\sqrt{a(a-1)} + (n-a)}{n} = \frac{a - \sqrt{a(a-1)}}{n}.$$

To derive a lower bound, we use the identity

$$a - \sqrt{a(a-1)} = \frac{a^2 - a(a-1)}{a + \sqrt{a(a-1)}} = \frac{a}{a + \sqrt{a(a-1)}}, \qquad a \geq 1.$$

Hence

$$H^2(P,Q) = \frac{1}{n} \cdot \frac{a}{a + \sqrt{a(a-1)}} \geq \frac{1}{n} \cdot \frac{a}{2a} = \frac{1}{2n},$$

since $a + \sqrt{a(a-1)} \leq 2a$. Taking square roots completes the proof:

$$H(P,Q) \geq \sqrt{\frac{1}{2n}} = \frac{1}{\sqrt{2n}}.$$

□

This lower bound is the same as that under the DELETE operation. Therefore, $\theta = 0.1$ enables Hellinger distance to identify disruption in extreme cases.

## References

[1] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2015. Personalized tour recommendation based on user interests and points of interest visit durations. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. 1778–1784.

[2] Cristina Ioana Muntean, Franco Maria Nardini, Fabrizio Silvestri, and Ranieri Baraglia. 2015. On learning prediction models for tourists paths. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7, 1 (2015), 1–34.

[3] Xiaoting Wang, Christopher Leckie, Jeffrey Chan, Kwan Hui Lim, and Tharshan Vaithianathan. 2016. Improving personalized trip recommendation by avoiding crowds. In *Proceedings of the 25th ACM international on Conference on Information and Knowledge Management*. 25–34.