

return_dim_tag_20251...
文件

return_fact_details_202...
文件

你是一名资深的亚马逊美国站点的产品总监，同时也是运营总监，你让一名运营小白根据return_dim_tag的标签定义和边界进行退货留言的打标，return_fact_details是运营小白交付的打标结果，现在请你对交付结果的准确度逐条进行打分

已思考 7m 19s >

已思考 1m 18s >


先给你一个整体结论，然后逐条打分。


评分规则（我这边统一定义一下）：

- **1 分**：完全符合这个维度的定义和边界，是“理想答案”
- **0.5 分**：方向大体对，但和定义的边界有偏差 / 有更合适的主标签
- **0 分**：不符合该标签定义，不推荐打这个标签

整体来看：

- **1 分：45 条**
- **0.5 分：5 条**
- **0 分：8 条**
- 严格按“完全正确=1”的准确率约 **78%**，把 0.5 也算作半对，整体有效率约 **82%**。

标签定义来自 return_dim_tag ，评分依据为定义与边界说明。  return_dim_tag_20251114

打标结果来自 return_fact_details 。  return_fact_details_20251114

逐条打分与点评（按文件顺序）

序号	review_id	tag_code	得分	简短点评
1	R1LTUUWSTZ9AG G	INSTALL_FIX	0.5	只抱怨“组装很烦”，没谈调好后易位移/需频繁紧固，更像“安装复杂”，和INSTALL_FIX 定义有偏。
2	R1O02YVMHQ8M NS	MATERIAL_RUST	1	“the screws are not rustproof” = 螺丝不防锈，完全符合“材质/防锈失效”。
3	R87ZB4398VG0G	STABILITY_TIP	1	“Cannot stand my dish” 理解为架不住盘子，对应“稳定性/防倾倒不足”合理。
4	RCYO6MIQT6IE1	SCENE_ADAPT	0.5	问题是“占满水槽右侧”，属于场景使用不理想，但未出现倾斜/下坠/回流水等典型故障，勉强算场景适配不足。
5	RCYO6MIQT6IE1	GUIDE_VARIANT	0.5	“I should have checked the specs.” 有误选成分，但用户也自认是自己没看清，是否归因于指引不清不确定。
6	R1CXWTGPPXE6R Q	PAGE_MISSING	1	“Was unaware it needed to be put together” = 不知道要组装，可视为页面对安装要点提示不够。
7	R1EUI60POM0L4D	MATERIAL_RUST	1	用了 5 个月生锈，典型材质/防锈失效。

序号	review_id	tag_code	得分	简短点评
8	R1FKKN3V47NYGP	SCENE_ADAPT	0	文案里是“fit almost any sink” 正向评价，不是“场景适配不足”。这个标签不该打。
9	R1FKKN3V47NYGP	STABILITY_TIP	1	“a little wobbly when stretching it out full length” 是晃动问题，完全符合稳定性不足。
10	R1GCBKRVQ36CPT	FIT_COMPAT	1	“did not fit” + 退货，典型“尺寸/兼容性不符”。
11	R1GCBKRVQ36CPT	GUIDE_VARIANT	0.5	“Much smaller than anticipated” 有预期管理/选型误差，可能和指引相关，但也可能是买家自己没看尺寸。
12	R1OTE4Q3RSD124	CAPACITY_EFF	1	“Muy pequeño/非常小”，直接指向容量/有效面积偏小，可视为容量不足。
13	R2CZ4AZ0E543P5	INSTALL_FIX	1	“The screws fell out numerous times” 需要不断处理，典型固定性差。
14	R2UQOC992TAEO7	CS_POOR	0	客服让寄回，用户抱怨“去 UPS 太麻烦”，更多是退换流程/自身不便，不是客服态度或处理不达标。
15	R2UQOC992TAEO7	INSTALL_FIX	1	用一个月螺丝开始脱落、要用铁丝绑住，完全是安装/固定性问题。

序号	review_id	tag_code	得分	简短点评
16	R38AZO0RUC1G17	PAGE_MISSING	1	"I did not realize that you would have to assemble once you received it." 典型"页面未清晰提示需要组装"。
17	R3H0XMUHSAO28	FIT_COMPAT	1	"too small for my kitchen sink" 明确尺寸不合。
18	R3H0XMUHSAO28	SCENE_ADAPT	0	根因是"太小"，按边界说明：尺寸不合应归FIT_COMPAT，不应再打SCENE_ADAPT。
19	R3J3G4WC0X1E1V	DOA_DAMAGED	1	出厂配件没装好，一部分橡胶件只是丢盒子里且无法自己安装 = 开箱即功能不完整，典型DOA/缺件。
20	R3J3G4WC0X1E1V	PARTS_DESIGN	0	问题核心是"少装/装错"，不是结构设计本身不合理，因此不建议打PARTS_DESIGN。
21	RI20ZLLKL9DYG	FIT_COMPAT	1	"Didnt fit my sink at all and I have a standard size sink." = 标称应适配标准水槽却没适配，完美命中。
22	R1P8UT2IK6F4E7	MATERIAL_RUST	1	两个月就生锈，典型防锈失效。
23	R26Z7K293JZOHP	STABILITY_TIP	1	"Flimsy" 可视为结构发软、不稳，归为稳定性不足合理。

序号	review_id	tag_code	得分	简短点评
24	R3T9650HZPHPC0	SOCIAL_GAP	0	“Not what I expected” 是预期落差，没有提高 评分/热卖等社会证明， 与定义不符。
25	R3T9650HZPHPC0	STABILITY_TIP	1	“not sturdy very wobble on top of sink 是非常典型的晃动/不稳 问题。
26	R1DIR5UMMQVM 8V	LOOK_COLOR	0	“its not stainless steel” 是材质/规格问题，不是 外观色差，应考虑 SPEC_MISMATCH 或材 质相关标签。
27	R1DIR5UMMQVM 8V	INSTALL_FIX	1	“little knobs ... move around” 位置锁不住， 属于固定性/调节性差。
28	R1DIR5UMMQVM 8V	PARTS_DESIGN	1	餐具筐浅、感觉廉价， 是典型“刀叉篮/杯架设计 不合理”。
29	R1DIR5UMMQVM 8V	VALUE_WEAK	1	“not worth what I paid”“should be \$10 or less” 直接表达价格价 值感不佳。
30	R1H5889KTI8IHE	MATERIAL_RUST	1	不到一年就生锈，符合 材质/防锈失效。
31	R18VMLQIBLIK2W	MATERIAL_RUST	1	用 3 天就生锈，更是典 型防锈失效场景。
32	R2AVPEX1OYOUY Q	INSTALL_FIX	1	需要不断重新拧螺丝、 复位橡胶件，完全符合 INSTALL_FIX 定义。

序号	review_id	tag_code	得分	简短点评
33	R2AVPEX1OYOUYQ	VALUE_WEAK	1	“会去找替代品” = 明确有换购倾向，符合VALUE_WEAK。
34	R2EKMFA0HY6EZ7	CAPACITY_EFF	1	“不能很好最大化空间” “像缩小版的大架子” = 有效面积不足。
35	R2EKMFA0HY6EZ7	PARTS_DESIGN	1	碗盘在排水区“坐不稳”，是分区结构导致的使用问题。
36	R2IJTUIDOBBGTR	FIT_COMPAT	0.5	“mostly fits... sides still open” 只是两侧有空，不是明显无法放置，勉强算兼容性略差。
37	R2IJTUIDOBBGTR	PARTS_DESIGN	1	餐具架过大，应更短才能稳固，是典型配件设计不合理。
38	R3441UHDAP5UL8	VALUE_WEAK	1	“看起来很低质又廉价，不喜欢” = 整体价值感偏低，归 VALUE_WEAK 合理。
39	R3GDDPAC4WALFE	INSTALL_FIX	0	问题是“需要组装”和“Flimsy”，对应PAGE_MISSING + STABILITY_TIP，更不是“调好后固定性差”。
40	R3GDDPAC4WALFE	SOCIAL_GAP	0	仅“Not what I expected”，同样没提社交证明，高评分/热卖等。
41	R3LFGFO6Z93AP5	SPEC_MISMATCH	1	“it said rust resistance...that’s why I bought it” 且实际生

序号	review_id	tag_code	得分	简短点评
				锈，规格宣传与实物不符。
42	R3LFGFO6Z93AP5	MATERIAL_RUST	1	“turns rusty bad” 是典型材质/防锈失效。
43	RD7VR09KC85WW	VALUE_WEAK	1	“Very cheaply made. Do not recommend” = 质量感差且明确不推荐，符合价值/对比劣势。
44	R2UPS2YYUD7M70	FIT_COMPAT	1	“rack didn’t fit my sink properly because my faucet... too wide” = 与实际水槽结构不兼容。
45	R2UPS2YYUD7M70	VALUE_WEAK	1	材料感觉便宜、质疑耐用性与价格匹配度，是典型价值感偏低。
46	R2UPS2YYUD7M70	STABILITY_TIP	1	“It’s also a bit flimsy... don’t feel confident with heavy dishes” 典型稳定性/承重可靠性不足。
47	R2Z4UK8H91SCGR	CAPACITY_EFF	1	伸展后有大缝，东西会掉下去 = 有效可用面积被浪费，容量效果不达标。
48	R2Z4UK8H91SCGR	PARTS_DESIGN	1	刀叉杯太小且太高挂不上，是典型配件几何/结构问题。
49	R3DK7EH9J86G2C	LOOK_COLOR	1	“a little different than the illustration shows” = 实物和图示不一致，归外观/颜色不符合理。

序号	review_id	tag_code	得分	简短点评
50	RG4YQ1P976ZJY	INSTALL_FIX	1	螺丝位置刚性、无法灵活调整外侧，属于安装/调节位置受限。
51	RGOK60EVGW06H	CAPACITY_EFF	1	“Not very big” 直指容量/有效面积偏小。
52	RGOK60EVGW06H	VALUE_WEAK	1	“Strangely made and very awkward. I dont recommend.” 虽未提价格，但整体体验差+明确不推荐，可归为价值/对比劣势。
53	R38XZO2C596REH	DOA_DAMAGED	1	缺所有连接件，无法使用，是标准缺件/DOA场景。
54	RO4T32WPBCGRD	PARTS_DESIGN	1	没有防倒的横杆导致高杯子容易倾倒，典型杯架结构设计缺失。
55	RO4T32WPBCGRD	STABILITY_TIP	1	“Cups fall over...” 是直接的防倾倒/稳定性问题。
56	R384TSBX2ZQOS	PARTS_DESIGN	1	刀叉杯因横杆阻挡卡不上，是配件设计问题。
57	R384TSBX2ZQOS	STABILITY_TIP	1	伸缩机构造成台阶，餐具“不能平放”，属于放置稳定性问题。
58	R384TSBX2ZQOS	DOA_DAMAGED	1	“doesn’t sit straight; bent out of shape” 说明到货即变形，不是正常结构，符合损坏到货。

总结&训练建议（给这位运营小伙伴的）

1. 容易混淆的几个标签：

- **SOCIAL_GAP**：必须出现“评分/热卖/大家都说好”等“社会证明”线索，否则只是“和预期不同”，不该用这个标签。
- **SCENE_ADAPT vs FIT_COMPAT**：如果根因是尺寸不合（太大/太小），优先 **FIT_COMPAT**；只有尺寸 OK，但在某场景会倾斜、回流水、下坠等，再用 **SCENE_ADAPT**。
- **INSTALL_FIX vs STABILITY_TIP / PAGE_MISSING**：
 - 只说“难装”“需要组装”→更接近 **PAGE_MISSING**（页面没写清）
 - 说“晃、软、flimsy、not sturdy”→优先 **STABILITY_TIP**
 - 说“螺丝总掉、需频繁拧紧、位置锁不住”→才是 **INSTALL_FIX**
- **LOOK_COLOR vs SPEC_MISMATCH / MATERIAL_RUST**：
 - “不是不锈钢/材质不对”→考虑 **SPEC_MISMATCH** 或 **MATERIAL_RUST**（如伴随生锈）
 - “颜色和图不一样/涂层不均”→才是 **LOOK_COLOR**

2. 做得不错的地方：

- 所有跟 **MATERIAL_RUST**、**DOA_DAMAGED**、**VALUE_WEAK**、**PARTS_DESIGN**、**STABILITY_TIP** 相关的大部分案例，判断都很稳。
- 对于多标签场景（同一条 review 打多个维度），整体覆盖较完整，很少漏掉重要维度。

如果你后面想，我也可以帮你把这些错误样本单独整理出来，当作运营培训题库，再配几道“纠错练习题”给新人用。

来源



假设把误差原因都归属于我们自己，请你根据评分结果（<1分），哪些标签的定义和边界需要进一步的完善，是否需要新增标签，但是新增和定义的时候注意全局的MECE原则

已思考 36s >

已停止思考 >

假设把误差原因都归属于我们自己，请你根据评分结果（<1分），哪些标签的定义和边界需要进一步的完善，是否需要新增标签，但是新增和定义的时候注意全局的MECE原则

已思考 7m 10s >

先把结论说清楚：

从所有打分 < 1 的样本看，**真正暴露问题的不是操作员，而是我们自己的标签设计在几个地方不够 MECE（边界有重叠/空白）。**

需要重点优化的标签（按优先级）：

INSTALL_FIX、SCENE_ADAPT、FIT_COMPAT、GUIDE_VARIANT、CS_POOR、SOCIAL_GAP、LOOK_COLOR、PARTS_DESIGN（和 DOA_DAMAGED 的边界）
建议**新增 1 个标签**：“**组装复杂/耗时高**”（暂名：INSTALL_COMPLEX）

下面我按“问题簇”来讲：哪里不 MECE、怎么改定义&边界、是否需要新标签。

一、安装相关：INSTALL_FIX 太“窄”，导致被乱用

涉及误差样本：

- “very frustrating to put together”（组装很烦）→ 被打成 INSTALL_FIX
- “Needed to be put together. Flimsy” → 被打 INSTALL_FIX，而不是 PAGE_MISSING + STABILITY_TIP

1) 现在的问题

- INSTALL_FIX 的定义是**“装配虽易，但调好后易位移、需要频繁紧固或无法锁定位置”**。
- 但评论里大量是：“**步骤多、螺丝多、说明难懂、自己装起来很痛苦**”，并没有说“调好后还老是松”。

也就是说：

“过程痛苦”和“结果锁不稳”目前被塞在一个标签里，而定义又写得很偏“结果”，操作员只能猜。

2) 建议：新增一个安装维度标签（必要）

新增标签（建议）：`INSTALL_COMPLEX`

- **tag_name_cn**: 组装复杂/耗时高
- **建议归类**: 也在结构/使用成本相关大类里（和 `INSTALL_FIX` 同一层级）
- **定义（草案）**:

“安装/组装步骤繁琐、耗时长、需要额外工具或说明书难懂，导致用户明显表达‘装起来很痛苦’。”

- **边界建议**:
 - 若抱怨是“装好了之后还总要拧螺丝/位置老是跑” → `INSTALL_FIX`
 - 若抱怨是“页面没说要自己组装，我以为是整装” → `PAGE_MISSING`
 - 若抱怨是“*Flimsy/not sturdy/wobbly*” → 优先 `STABILITY_TIP`，而不是 `INSTALL_FIX`

同时微调 `INSTALL_FIX` 定义：

加一句：“不包含‘仅因步骤多、说明不清导致安装过程挫败’的情况，这类归入‘组装复杂/耗时高’。”

这样安装相关就变成一个 MECE 的小树：

1. 页面没讲清要自己装 → `PAGE_MISSING`
2. 说明写了，但安装过程太折腾 → `INSTALL_COMPLEX`（新）
3. 装好后老是松/位置锁不住 → `INSTALL_FIX`
4. 结构本身软/晃/易倒 → `STABILITY_TIP`

二、适配&场景：FIT_COMPAT / SCENE_ADAPT / GUIDE_VARIANT 三兄弟边界不够锐

典型误差样本：

- 想要更长/更大：“I would have preferred that it could expand ... Completely on me though - I should have checked the specs.”

被打：`SCENE_ADAPT + GUIDE_VARIANT`

- “Much smaller than anticipated and did not fit. Returned.”
被打: **FIT_COMPAT + GUIDE_VARIANT**
- “too small for my kitchen sink / smaller than my RV sink”
同时被打: **FIT_COMPAT + SCENE_ADAPT**

1) 现在的问题

- **FIT_COMPAT** 定义里写“在标称适配范围内仍无法稳固放置”，容易让人误解成：
“评论里没提‘页面写了适配范围’，就不敢打 **FIT_COMPAT**。”
- **SCENE_ADAPT** 定义是“某/多场景下出现倾斜、下坠、回流水等”，但实际使用时也被用在纯尺寸问题上（太小/太大）；
- **GUIDE_VARIANT** 的定义是“说明不足导致误选”，但样本里很多只是“我主观以为会更大/更长”，并未说“页面不清楚”。

2) 建议：优化这三个的定义&边界（不一定要新标签）

(1) 放宽 **FIT_COMPAT** 的前提表述

调整 **FIT_COMPAT** 定义为：

“实际与用户目标位置/设备（如水槽尺寸、柜体尺寸）不匹配，导致**无法放入/无法跨放，或放入后间隙过大严重影响使用。**

不要求评论里必须提‘页面宣称适配’。”

边界补充一句：

- 若只是“放得下，但某个姿势略有不便”，无明显功能问题 → 不打 **FIT_COMPAT**，按照 **SCENE_ADAPT** 或其他来。

(2) 收紧 **SCENE_ADAPT**

强调：

“**SCENE_ADAPT** 仅用于：尺寸本身能放下，但在特定场景（双水槽、RV、窄台面）下出现倾斜/下坠/回流水等现象。

根原因为尺寸完全不合的情况 → 一律 **FIT_COMPAT**。”

(3) 收紧 **GUIDE_VARIANT** 应用条件

在定义和边界里加“必要条件”：

- “出现类似字眼：“**listing confusing / pictures misleading / sizes all on one page not clear / hard to tell which size to choose / should be clearer which size fits**”
- “或用户明确把责任归因于**页面/卖家指引**，而不是“完全怪我自己”。”

并在边界里写清楚：

- 只说 “smaller than expected / I should have checked the specs” → **默认不打 GUIDE_VARIANT，仅打 FIT_COMPAT。**

这样三者就更接近 MECE：

- 完全放不下/明显尺寸不合 → FIT_COMPAT
- 尺寸 OK，但在某特殊场景下有倾斜/回流 → SCENE_ADAPT
- 多规格/多尺寸商品，页面指引混乱 → GUIDE_VARIANT

三、服务/退货：CS_POOR vs RETURN_POLICY_BAD 分工不清

典型误差样本：

- “客服让寄回，但 UPS 很远，我懒得去” → 被打成 CS_POOR

1) 现在的问题

- CS_POOR（品牌/店铺服务不达标）定义偏客服响应/态度；
- 但样本里抱怨的是：**退货渠道对偏远用户不友好（要开车去 UPS，很麻烦）**——更像退换流程/政策设计问题。

2) 建议：不新增标签，强化边界

强化 RETURN_POLICY_BAD **定义：**

明确包含：

- “退货地点选择少、离家太远”
- “必须指定快递点（如 UPS）对某些地区非常不方便”
- “包装、打印标签、预约取件等流程复杂”

同时把 CS_POOR **边界写死：**

仅当用户抱怨：

- “客服慢、不回消息”
- “承诺的补偿/换货没有兑现”
- “给错信息/推诿等

才打 CS_POOR。

对于“流程设计本身对我不友好”，一律归 RETURN_POLICY_BAD。”

四、感知类：SOCIAL_GAP 的“社会证明”要写死，不然容易被当成“预期落差”

典型误差样本：

- “Not what I expected.”
- “I thought it would be a little bit different...”

都被打成 SOCIAL_GAP，但评论里完全没有提“高评分/热卖/好评”等社会证明。

1) 现在的问题

SOCIAL_GAP 目前定义：

“热卖/高评等社证与实际体验差距大，削弱信任。”

但实际操作中，“Not what I expected”太像“预期落差”，容易被误扔到这里。

2) 建议：不新增“预期落差”标签，只把 SOCIAL_GAP 收紧

- 在定义中加一句“必须出现社会证明相关线索：如‘评分 4.8 却很一般’、‘评论都说很好结果一般’、‘bestseller 但体验不如其他’等”。
- 在边界写明：

仅有 “not what I expected / I thought it would be different” 且无法从文本推断 “是被评分/评论影响预期”的，**不打 SOCIAL_GAP**，只按具体问题打功能标签（如 STABILITY_TIP、CAPACITY_EFF 等），没有就可以不打“体验维度”的标签。

这样避免新增一个非常模糊、极易跟所有维度重叠的“预期落差”标签，保持 MECE。

五、外观/材质：LOOK_COLOR 与 SPEC_MISMATCH 的边界不清晰

典型误差样本：

- “Its sturdy but its not stainless steel.” → 被打成 LOOK_COLOR

1) 现在的问题

- LOOK_COLOR 目前是**“色差/涂层不均/边角发灰”**，本质是“视觉效果与图不符”；
- 而“不锈钢 / 材质”其实是**客观规格**，更接近 SPEC_MISMATCH；但 SPEC_MISMATCH 定义现在偏“伸缩范围/承重/重量/容量等硬指标”。

2) 建议：扩大 SPEC_MISMATCH 范围，不新增新标签

修改 SPEC_MISMATCH 定义：

“伸缩范围、承重、重量、容量、材质（如是否不锈钢）、是否可洗碗机等**客观规格**与页面宣称不符。”

在 LOOK_COLOR 边界加：

- “若投诉点是“材质/是否不锈钢/是否金属/是否塑料” → **SPEC_MISMATCH**”
- “LOOK_COLOR 仅用于“颜色/光泽/图案/视觉效果”的不一致。”

这样“外观色差”和“材质规格不符”就被拆开，互斥清晰。

六、设计 vs 质量：PARTS_DESIGN 与 DOA_DAMAGED 的拆分

典型误差样本：

- 出厂没把橡胶件装好，用户自己装不上：“It is something that has to be done during manufacture.”
→ 被打：DOA_DAMAGED + PARTS_DESIGN

1) 现在的问题

- PARTS_DESIGN 定义是真正的“结构几何问题”（太高/太窄/缺防倾结构）；
- 上述案例其实是“生产装配缺失/导致到货即不可用”，**本质是 DOA**，不是“设计不合理”。

2) 建议：不新增标签，只补充 DOA 的边界

在 DOA_DAMAGED 边界里明确写入：

- “若用户描述“某个配件原本应该在出厂时装好/焊好/固定好，但现在是松的/没装/装错，导致开箱即无法正常安装或使用”，
→ 归 **DOA_DAMAGED**，不再打 PARTS_DESIGN。”

PARTS_DESIGN 保持只关注：

- 形状/高度/宽度/孔位导致杯子/刀叉放不上、太浅、易倒等纯“几何结构”问题。

七、FIT_COMPAT 的“半对样本”：什么时候不该打？

有一条是：

“Mostly fits the sink, but there is still space left open on the sides... I understand

it won't be perfect because this depends on the sink.”

被打上了 FIT_COMPAT + PARTS_DESIGN。

从 MECE 角度看：

- 他自己说 “I understand it won't be perfect” → 说明“侧面有空隙”是**可接受范围内的兼容性缺点**，且没有说因此导致严重功能问题。
- 真正的痛点在“utensil holder too large to fit well / should be shorter so it stays in place” → 这是很典型的 PARTS_DESIGN。

建议在 FIT_COMPAT 边界加一条：

- “若仅为“轻微缝隙/不完美贴合”，且用户主要抱怨集中在某个配件设计上（如餐具篮尺寸不对），
→ 优先打 PARTS_DESIGN，不打 FIT_COMPAT，避免重复和泛化。”

总结：要改的 & 要新增的

1. 建议重点修改定义/边界的标签

- **INSTALL_FIX**：和安装过程/稳定性要重新划清界线
- **FIT_COMPAT**、**SCENE_ADAPT**、**GUIDE_VARIANT**：三者之间的分工要写得更“像决策树”
- **CS_POOR** vs **RETURN_POLICY_BAD**：谁管流程、谁管客服，边界要写死
- **SOCIAL_GAP**：必须出现“社会证明”线索，否则不打
- **LOOK_COLOR** vs **SPEC_MISMATCH**：材质/是否不锈钢统一纳入 SPEC_MISMATCH
- **PARTS_DESIGN** vs **DOA_DAMAGED**：出厂装配缺失一律 DOA，不算设计问题
- **FIT_COMPAT** 的适用条件：弱问题不打，避免到处挂 FIT_COMPAT

2. 建议新增 1 个标签（保持全局 MECE，又不过度拆分）

- **INSTALL_COMPLEX**（组装复杂/耗时高）
 - 补上“仅抱怨安装过程痛苦”这块空白，避免大家把所有“安装相关抱怨”都塞进 INSTALL_FIX 或 PAGE_MISSING 里。

如果你愿意，下一步我可以帮你把这些改动整理成一份***打标决策树 + 反例说明***，方便你直接拿去培训运营同学。



return_dim_tag_20251...
文件

附件是v1版本的标签库，请你在v1的基础上对标签定义和边界进行优化，被改动标签对应的version改成v2，最后生成JSON文件供我下载

已思考 28s >

已思考 1m 31s >

我已经在你给的 v1 标签库基础上，按照前面讨论的 MECE 原则做了这些调整，并生成了一个新的 JSON 文件：

👉 下载链接：

[return_dim_tag_v2_20251117.json](#)

本次具体改动一览（均已在 JSON 中落地）

一、新增标签（version=1）

1. INSTALL_COMPLEX 安装/组装复杂（耗时高）

- **definition:**

安装/组装步骤繁琐、耗时明显偏长，需额外工具或说明难以理解，用户明确表达“装起来很痛苦/费劲”等感受。

- **boundary_note:**

若已安装完成且主要问题为螺丝/卡扣反复松动或位置锁不住→“安装/调节/固定性差”；仅因页面未说明需要自行组装导致心理落差→“页面信息缺失/误导”。

用于承接“安装过程很烦/很难”这类抱怨，和 INSTALL_FIX 做拆分。

二、修改且 version=2 的标签

以下标签的 `version` 已全部从 **1 改为 2**，并更新了 `definition / boundary_note` 以及 `updated_at`：

1. **FIT_COMPAT** 尺寸/兼容性不符
2. **SCENE_ADAPT** 场景适配不足
3. **GUIDE_VARIANT** 变体/选型指引不清
4. **CS_POOR** 品牌/店铺服务不达标
5. **RETURN_POLICY_BAD** 退换信息/政策体验不佳
6. **SOCIAL_GAP** 社会证明与体验不符
7. **SPEC_MISMATCH** 规格参数与宣传不一致
8. **LOOK_COLOR** 外观/颜色与实物不符
9. **INSTALL_FIX** 安装/调节/固定性差
10. **DOA_DAMAGED** 损坏到货 (DOA/受损)
11. **PARTS_DESIGN** 分区/配件设计不合理
12. **PAGE_MISSING** 页面信息缺失/误导

这些改动的核心逻辑是：

- **安装相关：**
 - **INSTALL_COMPLEX** 专门接住“安装过程复杂/耗时高”；
 - **INSTALL_FIX** 只管“装完后还锁不住/总要拧螺丝”；
 - 和 **STABILITY_TIP**、**PAGE_MISSING** 的边界在文案里都写清楚了。
- **适配相关 (FIT_COMPAT / SCENE_ADAPT / GUIDE_VARIANT)：**
 - **FIT_COMPAT**：只要是**尺寸不匹配导致基本放不下/缝隙巨大**就归它，不再强依赖“标称适配范围”这句话；
 - **SCENE_ADAPT**：强调“尺寸基本 OK，但在 RV/双槽/窄台面等特定场景出现倾斜/回流”；
 - **GUIDE_VARIANT**：必须能看出用户把“选错型号”归因到**页面指引不清**，而不是单纯“我没看清”。
- **服务/退货分工 (CS_POOR vs RETURN_POLICY_BAD)：**
 - **CS_POOR**：只认客服响应/态度/个案处理；
 - **RETURN_POLICY_BAD**：流程复杂、必须跑远处 UPS 才能退、规则不友好等，全部归它。
- **预期 vs 社会证明 (SOCIAL_GAP)：**
 - 必须出现“高评分/热卖/大家都说好”这类**社会证明**字眼才打；
 - 仅“not what I expected”不再归这里。
- **外观 vs 规格 (LOOK_COLOR vs SPEC_MISMATCH)：**

- 材质/是否不锈钢/是否塑料等 → SPEC_MISMATCH;
- 真正的“色差/光泽度/涂层不均”等 → LOOK_COLOR。
- **设计 vs 出厂缺陷 (PARTS_DESIGN vs DOA_DAMAGED) :**
 - 几何/结构设计问题 → PARTS_DESIGN;
 - 出厂装错/少装导致开箱就不能正常安装/使用 → DOA_DAMAGED。
- **信息缺失 (PAGE_MISSING) :**
 - 明确把“未说明需要自行组装”写进了定义;
 - 若信息已清晰写出, 仅是用户没仔细看 → 不打 PAGE_MISSING。

你可以直接下载上面的 `return_dim_tag_v2_20251117.json` 替换到你的系统里使用。
如果后面你想, 我也可以基于这套 v2 标签库帮你生成一份**「运营打标训练题 + 答案」**
**, 方便培训新人。 </>

📄 🍷 🗨️ ⬆️ ↺️ ...

请你重新审视V2版本, 整体来看, 是否有删减标签的空间

已思考 1m 16s >

有空间, 但不需要“大动刀子”。

从现在这套 V2 来看, **整体已经比较 MECE**, 但如果你的核心诉求是:

「一线运营更好上手、单条 review 打标更快」,
那确实有几类标签可以**合并/下沉成“衍生维度”**, 不必让新人手动选。

我按「推荐保留 / 建议合并或删除」给你一个清单。

一、我建议坚决保留的“主干维度” (不要动)

这些是你现在这套体系里, **高频 + 高价值 + 边界清晰**的标签, 不建议删减:

- **适配&场景类**
 - FIT_COMPAT 尺寸/兼容性不符

- SCENE_ADAPT 场景适配不足
- GUIDE_VARIANT 变体/选型指引不清
- **页面&规格类**
 - PAGE_MISSING 页面信息缺失/误导
 - SPEC_MISMATCH 规格参数与宣传不一致
- **安装&使用类**
 - INSTALL_COMPLEX 安装/组装复杂（耗时高）
 - INSTALL_FIX 安装/调节/固定性差
 - STABILITY_TIP 稳定性/防倾倒不足
 - CAPACITY_EFF 容量/有效面积不足
 - PARTS_DESIGN 分区/配件设计不合理
 - CLEAN_MAINT 清洁/维护成本高
- **产品本体质量类**
 - DOA_DAMAGED 损坏到货（DOA/受损）
 - MATERIAL_RUST 材质/防锈失效
- **物流&售后类**
 - SLA_DELAY 物流时效异常
 - RETURN_POLICY_BAD 退换信息/政策体验不佳
 - CS_POOR 品牌/店铺服务不达标
- **价格&感知价值类**
 - VALUE_WEAK 价值/对比劣势

这些构成了“问题是什么”的主干树，删掉会明显损失分析颗粒度，我不建议动。

二、明确有删减/合并空间的标签（重点看这几个）

1) SOCIAL_GAP 社会证明与体验不符 → 建议并入 VALUE_WEAK

现状：

- 使用条件苛刻：必须同时出现“高评分/热卖/大家都说好”+“体验不如预期”。
- 在实际样本里出现频率不会太高，而且它往往**同时伴随其他真实问题**（如质量差、稳定性差）。

问题：

它本质是一个“情绪 + 信任”维度，而不是独立的“问题类型”。

从运营决策角度看：

- 把它单独做一个标签,
- 和“到底哪里不好”那一堆标签并列,
收益有限, 认知负担反而上升。

建议:

- **删除** SOCIAL_GAP **这个标签;**
- 逻辑上并入:
 - 用户觉得被“评分/评论骗了” → 依然靠
 - **根因标签** (如 STABILITY_TIP、MATERIAL_RUST、PAGE_MISSING 等)
 - **价值维度**: VALUE_WEAK 来体现负面体验。

简单说:

“社会证明 ≠ 实际体验” 这个信息, 可以通过

「高星 + 吐槽 + VALUE_WEAK」

在后续分析阶段算出来, 没必要让一线手动打一个专门标签。

2) PRICE_PROMO_OPAQ 价格/促销不透明 → 建议合并到 VALUE_WEAK / RETURN_POLICY_BAD

现状:

- 覆盖“价格波动/买贵了/活动解释不清/价格很套路”这类反馈。
- 实战上, 抱怨价格不透明的评论, **几乎总是伴随着“觉得亏/不值”。**

问题:

- 它和 VALUE_WEAK 的边界非常“概念化”, 一线打标很难稳定区分:
 - “觉得卖得贵” → VALUE_WEAK
 - “觉得被活动套路了” → PRICE_PROMO_OPAQ但在用户嘴里往往混在一起。

建议:

- **删除** PRICE_PROMO_OPAQ **作为独立标签;**
- 处理方式:
 - 纯“贵、不值” → 仍然用 VALUE_WEAK
 - 明确指向“退换差价、活动规则坑、退款不到位”等 → 用 RETURN_POLICY_BAD (退换/政策体验)
- 后期如果想看“价格相关问题”, 可以通过关键词 (price / deal / discount / coupon / promotion) 和 VALUE_WEAK / RETURN_POLICY_BAD 组合筛出来, 无需多一个标签。

3) SURFACE_PROTECT 台面/器具保护不足 → 建议合并进 PARTS_DESIGN / DOA_DAMAGED

现状：

- 专门描述“没有防滑/防刮脚垫、划伤台面/器具”等。
- 但本质上是**设计或装配问题**的一种表现形式。

问题：

- 与 PARTS_DESIGN 、 DOA_DAMAGED 高度重叠：
 - 如果本来就该有橡胶脚垫但没有设计 → PARTS_DESIGN
 - 如果说明书/图示里有脚垫，但实际到货缺失或装错 → DOA_DAMAGED

建议：

- **删除 SURFACE_PROTECT 单独标签；**
- 在 PARTS_DESIGN 的例子补一句：

“包含因缺少防滑/防刮脚垫导致划伤台面或器具的情况（设计缺失）；”
- 在 DOA_DAMAGED 边界里保留现在这句：

“出厂应预装的胶脚/防滑件未装或装错，开箱即无法按正常形态使用。”

这样既不损失信息，又能让打标决策更简单：

划伤台面？→ 先看是“设计没做”，还是“配件少装/装错”。

三、争议较小但可以视业务情况再简化的标签

下面这几个**不是必须删**，但如果你未来要进一步压缩标签数，可以当作“第二批候选”。

4) WEIGHT_MOVE 自重过大/移动不便

- 对当前这类水槽架产品来说，**出现频率不高**；
- 但在更大件的品类（比如带水箱、带轮、金属家具类）会很有价值。

建议：

- 如果你的标签体系会复用到更多大件品类 → **保留**；
- 如果从战略上只打算服务“轻量级厨卫收纳”等品类 → 可以考虑未来把它合并进一个更大的“使用便利性”类（比如和 CLEAN_MAINT 一起，做一个大类维度，在上层分析做拆分，而不是一线打两个小标签）。

5) PACK_WEAK 包装防护不足

- 和 DOA_DAMAGED 的关系是：
 - PACK_WEAK = 保护手段弱（但商品可能完好）；
 - DOA = 结果坏掉了。
- 对供应链/仓储优化很有价值。

从 MECE 看，两者是“原因 vs 结果”，并不重合。

建议：

- 如果你有打算把“包装改善”单独作为一个项目（比如换填充物、改纸箱强度）→ 保留；
- 如果你只是想简化运营打标，而后续诊断就不会深入到“包装 vs 质检”层面 → 可以考虑只留 DOA_DAMAGED，把 PACK_WEAK 作为后续用文本规则自动判断的“隐形特征”，不让运营手动选。

四、整体删减建议小结

在不牺牲核心洞察的前提下，我会给你两档方案：

方案 A：轻度精简（推荐）

- 删除 3 个标签：
 - SOCIAL_GAP
 - PRICE_PROMO_OPAQ
 - SURFACE_PROTECT
- 其它标签保持不变。
- 总数从 23 → 20 个，一线认知负担下降明显，但维度结构还是很完整。

方案 B：更激进一点（视业务选择）

在方案 A 基础上，再视品类范围决定是否：

- 合并/隐藏 WEIGHT_MOVE（并入“使用便利性”大维度，在后台做文本识别）；
- 合并/隐藏 PACK_WEAK（把包装问题仅通过 DOA_DAMAGED + 关键字分析体现）。

如果你愿意，下一步我可以帮你：

- 按方案 A 生成一版精简后的标签 JSON（自动调好 version / 边界说明），

- 顺便输出一张「老标签 → 新标签」映射表，方便你做历史数据迁移。



按方案 A 生成一版精简后的标签 JSON（自动调好 version / 边界说明），此外，在调整的过程中，可以思考一下标签中文名称与对应定义的表达，目的是让运营人员看到标签名称就大概猜到定义是什么

已思考 28s >

已思考 40s >

已思考 1m 8s >

已思考 31s >

我已经按**方案 A**帮你生成了一版精简后的标签库 JSON，并在调整过程中顺带优化了部分中文标签名，让运营同学一看名字大概就能猜到含义。

👉 下载文件：

[return_dim_tag_v2_slim_20251117.json](#)

本次精简 & 调整概览

一、删除了 3 个标签（不再出现在 JSON 中）

1. SOCIAL_GAP
2. PRICE_PROMO_OPAQ
3. SURFACE_PROTECT

对应的信息，会通过组合其它标签 + 文本分析在后续分析层面还原，不再让一线手动选。

二、中文名称做了“小手术”的标签（更贴近定义）

这些标签的 **tag_code** 不变，只是中文名字更口语直观：

1. PACK_WEAK

- **原名：** 包装防护不足
- **新名：** **包装太薄/防护不足**
- 运营看到就知道是“包装太单薄、容易磕碰”的那种吐槽。

2. VALUE_WEAK

- **原名：** 价值/对比劣势
- **新名：** **性价比差/不值这个价**
- 明确就是“贵”“不划算”“不值”的场景。

3. STABILITY_TIP

- **原名：** 稳定性/防倾倒不足
- **新名：** **整体稳定性差/易晃动**
- 一眼就联想到“晃、软、摇、容易倒”。

其余大部分标签中文名本身已经比较直观，我就没有硬改，比如：

- FIT_COMPAT：尺寸/兼容性不符
- GUIDE_VARIANT：变体/选型指引不清
- SCENE_ADAPT：场景适配不足（槽内/跨槽/台面/RV 等）
- INSTALL_FIX：安装/调节/固定性差
- CAPACITY_EFF：容量/有效面积不足（含扩容效果不达标）等。

三、定义 & 边界更新并升版（version=2）的标签

以下这些标签，**定义/边界有改动**，version 已统一从 1 升到 2，方便你在系统里区分 v1/v2：

- FIT_COMPAT 尺寸/兼容性不符
- SCENE_ADAPT 场景适配不足
- GUIDE_VARIANT 变体/选型指引不清
- CS_POOR 品牌/店铺服务不达标
- RETURN_POLICY_BAD 退换信息/政策体验不佳
- SPEC_MISMATCH 规格参数与宣传不一致（已纳入“材质是否不锈钢”等客观规格）
- LOOK_COLOR 外观/颜色与实物不符（只管“看起来不对”，材质不对归 SPEC_MISMATCH）
- INSTALL_FIX 安装/调节/固定性差（明确和“安装复杂”的拆分）

- DOA_DAMAGED 损坏到货 (DOA/受损) (补充了“出厂少装/装错配件”的场景)
- PARTS_DESIGN 分区/配件设计不合理 (与 DOA、稳定性拆分更清楚)
- PAGE_MISSING 页面信息缺失/误导
- PACK_WEAK 包装太薄/防护不足
- VALUE_WEAK 性价比差/不值这个价
- STABILITY_TIP 整体稳定性差/易晃动

整体思路就是：

- 文字里直接点出**常见话术** (比如“贵/不值/不划算”，“晃、摇、容易倒”，“包装太薄”)
- 边界里尽量写成**“如果是 A 场景就打 X，如果是 B 场景就打 Y”**这种运营能直接照抄的决策规则。

四、新增标签 (仍为 version=1)

保留之前加的安装维度标签：

- INSTALL_COMPLEX：**安装/组装复杂 (耗时高)**
 - 独立承接“步骤多、很难装、装到崩溃”这类吐槽；
 - 定义和边界里已经写清楚：
 - 只要是“过程很痛苦” → INSTALL_COMPLEX
 - “装好后总要拧螺丝/锁不住” → INSTALL_FIX
 - “根本没说要自己装” → PAGE_MISSING

五、保持不变的标签 (version 仍为 1)

这些标签逻辑上已经够清晰，这次没动定义与名称：

- SLA_DELAY 物流时效异常
- CLEAN_MAINT 清洁/维护成本高
- MATERIAL_RUST 材质/防锈失效
- CAPACITY_EFF 容量/有效面积不足 (含扩容效果不达标)
- WEIGHT_MOVE 自重过大/移动不便

如果你后面希望，我可以基于这份 **v2_slim 标签库** 再帮你出一份：

- 每个标签 2-3 条**典型好样本 + 易错反例**

- 外加一张「老标签 → 新标签（含删除/合并逻辑）」映射表

方便你直接拿去做**运营打标培训 PPT 或文档**。 </>



你觉得“场景适配不足（槽内/跨槽/台面/RV 等）”的标签名称是否可改成更直白点，倾向于“沥水”

已思考 30s >

我也觉得现在这个名字有点“产品经理味儿”，不够一眼秒懂 😊。

但如果直接改成“沥水问题”，会有点**缩窄了范围**——因为这个标签本来不仅管“水怎么流”，还管“在 RV、双槽、台面这种场景下的使用不顺”。

所以我会这样处理：**把“沥水”放到前面当记忆锚点，但保留“场景”两个字**，既直白又不丢边界。

建议的改名方案（推荐用第 1 个）

✅ 推荐名称：

「**沥水/场景不适配（槽内/跨槽/RV 等）**」

- 一线看到***“沥水”***：会想到回流水、积水、滴到台面这些典型抱怨；
- 看到***“场景不适配”***：还能 cover 掉
 - RV 里高度不合
 - 双水槽跨不过去
 - 台面太窄导致一边悬空

这类非“水路问题”但也是“场景相关”的吐槽。

你的 SCENE_ADAPT 定义本来就是：

在尺寸基本适配的前提下，在特定场景（双槽、RV、窄台面等）出现倾斜、下坠、回流水等问题

所以用「沥水/场景不适配」刚好把核心关键词都挂在名字里，运营一看就大概知道：

“不是单纯尺寸不合，而是放到某种场景里沥水/姿态出了问题”。

备选两个你可以对比看看

1. 「沥水不顺/场景不适配」

- 更口语一点，但“沥水不顺”略长，UI 上可能有点挤。

2. 「水槽场景不适配（沥水/倾斜等）」

- 强调是“水槽相关场景”，但如果你后面把这套标签复用到非水槽类品（比如台面置物架），就有点局限了。

如果你觉得「沥水/场景不适配（槽内/跨槽/RV 等）」OK，

下一步我可以帮你把 JSON 里 SCENE_ADAPT 的 tag_name_cn 更新成这个版本，并重新导出一版。



我觉得倾斜，下坠问题在其它标签已能体现出来，可以把这个标签的定义收窄为沥水相关的问题

好，这个收窄思路我很赞同：

“场景适配不足”→ 改成只管“水往哪儿流、流得顺不顺”，其它倾斜/下坠类都交给稳定性、尺寸这些标签。

我已经在你上一版精简 JSON 的基础上，完成了以下调整：

1. SCENE_ADAPT 标签改名 & 收窄定义

tag_code 不变： SCENE_ADAPT

- 新中文名称：

👉 「沥水/排水问题」

- 新定义 (definition)：

沥水方向或排水路径设计不佳，导致水流不到位或去向异常，如滴到台面/地面、流不到水槽、易积水不易干等。

- **新边界说明 (boundary_note) :**

若主要问题为架体在特定场景下倾斜、下坠或一侧悬空→优先按“整体稳定性差/易晃动”或“尺寸/兼容性不符”打标；

若仅为局部挡水条/接水盘几何设计导致局部不稳或空间浪费→可考虑“分区/配件设计不合理”；

本标签仅聚焦水路/沥水方向相关问题（如水滴外流、积水难排、滴在台面上等）。

- **version:** 已确保至少为 **2**，方便你在系统里区分新版定义。
- `updated_at` 也同步更新为当前时间。

这样一来：

- 各种“晃、倾斜、下坠、一边悬空” → 都去 `STABILITY_TIP`（整体稳定性差/易晃动）或 `FIT_COMPAT`；
- “沥水方向不对、滴到台面、积水不干” → 一律用 `SCENE_ADAPT` = 沥水/排水问题，一线看到名字就知道是水路问题。

2. 新文件下载

已经生成了新的 JSON 文件，供你直接使用：

👉 下载：

[return_dim_tag_v2_slim_drain_20251117.json](#)

如果你愿意，下一步我可以帮你写一小段“运营使用说明”，专门讲清楚：

- **沥水/排水问题 vs 整体稳定性差/易晃动 vs 尺寸/兼容性不符**
用几句“傻瓜式记忆口令 + 示例句”，方便培训新人。 </>

📄 🍷 🗨️ ⬆️ ↺️ ...

另外，这套标签体系中，与页面描述不符相关的我觉得可以酌情删除，因为后续我会再做依次页面描述匹配来做评估，

不在这里做打标，但是你要区别于期望落差相关的，期望落差还是需要保留的

已思考 38s >

已思考 31s >

已思考 31s >

理解你的思路了：

- “**页面描述是不是对的**” → 以后用独立的页面描述匹配体系来评估（自动化）；
- “**用户的主观预期 vs 实际体验**” → 仍然希望在这套人工标签体系里保留一个维度来抓。

我已经按这个原则，把标签做了一轮结构调整，并生成了一份新的 JSON 文件。

1. 做了哪些结构调整？

① 删除了“页面描述不符”类标签（从打标体系中拿掉）

从当前这版中 **移除** 了这两个标签（不再给运营打标用）：

1. SPEC_MISMATCH：规格参数与宣传不一致
2. LOOK_COLOR：外观/颜色与实物不符

它们都是典型的“**实物 vs 页面描述/图片不一致**”类型，后续交给你的“页面描述匹配评估”模块来做就好。

注意：只是**不让运营手动打这两标签**，概念和问题本身你还能通过别的算法去捕获。

② 保留但重写“信息类”标签的角色

保留了这两个：

- PAGE_MISSING 页面信息缺失/误导
- GUIDE_VARIANT 变体/选型指引不清

并重新明确它们的定位：

- **PAGE_MISSING**：只管“该说的没说清/没说全”（缺关键尺寸、缺安装说明、没讲要自己组装等等）；
 - 新边界里写清楚：
 - 页面数值错误、图片失真这类**客观描述不符** → 不再靠打标，交给页面匹配评估；
 - 用户只是没仔细看/预期太高 → 可考虑打“预期落差”，**不要打** PAGE_MISSING。
- **GUIDE_VARIANT**：只管“多尺寸/多版本下的指引不清”，例如：
 - “哪个尺寸适合我的水槽看不出来”
 - “图片/选项里看不明白差异”等
 - 边界中也删除了对 SPEC_MISMATCH 的依赖，改成：
 - 规格数值错误这类客观“描述不符” → 后台页面匹配模块处理。

③ 新增一个“预期落差”标签，专门抓用户主观期待

新增标签：

- EXPECT_GAP：**预期落差/没有想象中好**

字段设置：

- category_code：CAT_INFO_EXPECT
- category_name_cn：信息与预期管理
- version：1（新标签）

定义 (definition)：

页面信息本身未必错误，但用户主观预期（如尺寸、质感、整体体验等）明显高于实际表现，
在评论中使用“没有想象中那么好/不如预期/Not what I expected/Smaller than I expected”等表达。

边界 (boundary_note)：

- “若能明确指向具体功能/质量缺陷（如不稳、易生锈、容量太小等），应优先打对应功能性标签，可视情况叠加本标签；”
- “单纯认为“贵/不值”且与价格强相关→“性价比差/不值这个价”；”
- “因页面规格错误、图片失真等导致的客观“描述不符”问题，将在后续页面匹配评估中处理，本标签不再区分此类客观错误。”

也就是说：

- 用户说：“**Not what I expected / smaller than I expected / not as nice as I thought**”
 - 没有明显指向“页面写错、图骗人”等 → 打 EXPECT_GAP 。
- 如果他既说“没达到我的预期”，又具体说“很晃、会生锈”
 - 先打稳定性/材质等功能标签，**可选**叠加一枚 EXPECT_GAP 。

2. 同步调整了其他标签的边界描述（避免提到已删除的标签）

为了不让运营看到“指到一个不存在的标签”，我也顺手改了几处边界说明：

- FIT_COMPAT **尺寸/兼容性不符**
 - 去掉“→ 规格参数与宣传不一致”的引用，改成：
 - 由页面规格错误引起的认知偏差 → 由页面描述评估模块处理，本标签不管这块。
- GUIDE_VARIANT **变体/选型指引不清**
 - 边界中不再出现 SPEC_MISMATCH；
 - 增加：
 - “仅是自己没仔细看/预期更大，但没怪页面” → 用 FIT_COMPAT 或 EXPECT_GAP 。
- MATERIAL_RUST **材质/防锈失效**
 - 边界中不再提 LOOK_COLOR，改成泛化描述：
 - 单纯色差/喷涂不均等外观问题 → 交给视觉/页面质量评估，不用在这里打。
- PAGE_MISSING **页面信息缺失/误导**
 - 边界强调：
 - “页面没错，只是用户预期太高或没仔细看” → 打 EXPECT_GAP 不打 PAGE_MISSING；
 - 页面数值/描述错误 → 页面匹配模块处理，不再靠打标。

3. 新文件下载

我已经把这些变更写入一份新的 JSON 文件：

👉 下载：

[return_dim_tag_v3_expect_20251117.json](#)

这一版的特点是：

- 不再要求运营区分各种“页面描述对不对”；
- 专注在“实际体验是什么” + “用户主观预期是否落差很大”；
- 便于后续你用算法独立评估“Listing 文案/图片”这条链路。

如果你愿意，下一步我可以帮你：

- 用几条真实/模拟的评论，示范：
 - 哪些只打功能标签
 - 哪些加上 EXPECT_GAP
 - 哪些原本会打 SPEC_MISMATCH/LOOK_COLOR，现在直接“跳过交给页面评估模块” </>

< 2/2 >      ...

PAGE_MISSING和GUIDE_VARIANT是否可以整合到一个标签中即可，另外，保留LOOK_COLOR，EXPECT_GAP是否可以当成一个大类放到category_code即可，因为具体是什么不如预期就得看细分标签了，基于这些修改，你可以重新审视调整标签大类的分类以及标签命名

已思考 50s >

已思考 1m 36s >

太好了，这一步基本把体系「收敛成纯问题维度 + 预期靠细分标签」了，我已经按你的思路做完调整，并生成了新 JSON。

👉 新文件下载：

[return_dim_tag_v4_mergedInfo_20251117.json](#)

下面是关键变更说明，方便你 review：

1. 合并 PAGE_MISSING & GUIDE_VARIANT → 一个信息/指引标签

保留 tag_code: PAGE_MISSING

(减少历史数据迁移成本)

- **新中文名称:**

「页面信息/选型指引不清」

- **新定义 (definition) :**

商品页面在尺寸、适配范围、安装要点 (如需自行组装)、使用场景或多尺寸/多版本差异等关键信息上缺失、表达不清或难以理解, 导致用户对是否适配、安装难度或版本选择产生误判。

- **新边界 (boundary_note) :**

核心规则写成运营可以直接照抄的判定逻辑:

- 只说 “没有想象中那么好/以为会更大”, 没有提「页面/选型看不懂」
→ 按具体功能 + 价格标签处理, **不打这个标签**
- 评论里直接吐槽:
“尺寸对比看不懂 / 列表看不出差别 / 不知道该选哪个尺寸/型号”
→ 打 **页面信息/选型指引不清**, 不用再单独分变体标签
- 规格数值写错、图片严重失真这类**客观描述不符**
→ 不在打标侧区分, 交给你后续的“页面描述匹配评估模块”
- 页面已经写得很清楚, 只是用户没看
→ 不打这个标签
- **所属大类 (category) :**
 - category_code : 仍用 CAT_INFO_EXPECT
 - category_name_cn : 改为 **「信息说明与选型指引」**

2. 删除 EXPECT_GAP 标签 (预期落差不再做成一个单独 tag)

- 删除了 tag: EXPECT_GAP
- 预期落差今后不当成“一个标签”, 而是:
 - 通过**具体问题标签 + 星级/情绪**来理解
 - 或在分析层面通过 **star、review 文本、功能标签组合** 去识别

换句话说:

运营只打“具体哪里不好”, 而不是打一个抽象的“预期落差”。

3. 保留并重构 LOOK_COLOR → 只管「到手实物外观不满意」

根据你“页面描述不符交给后续算法”的要求，我把 LOOK_COLOR 重新加入体系，但**完全脱敏于“与页面是否一致”**：

新/现 tag_code： LOOK_COLOR

- 中文名称：

- 👉 「外观颜色/做工不满意」

- 定义：

- 用户对到手实物的颜色、质感、涂层、焊点/焊缝等外观做工不满意，如明显色差、看起来很廉价、涂层不均、焊接粗糙等。

- 边界：

- 重点是“到手实物长得丑/看着廉价/做工糙”，
 - 无论页面图片是不是一样，这里都不去判断。
 - 耐久性问题：生锈、掉皮、涂层寿命差 → MATERIAL_RUST
 - 外观问题只是支撑“这价不值” → 可叠加 VALUE_WEAK
 - 与页面图片/描述是否一致 → 交给你的 **页面评估模块** 做，不靠这个标签区分。

- 归类：

- category_code：CAT_MATERIAL_LOOK
 - category_name_cn：**材质与外观**

4. 现在的标签大类 & 命名结构 (v4)

当前这版共有 **18 个标签**，大类和命名已经比较清晰，运营一眼看名字大致就知道干嘛的：

A. 结构与承载 (CAT_STRUCT_LOAD)

- CAPACITY_EFF：容量/有效面积不足（含扩容效果不达标）
- PARTS_DESIGN：分区/配件设计不合理（刀叉篮/杯架等）
- STABILITY_TIP：整体稳定性差/易晃动
- WEIGHT_MOVE：自重过大/移动不便

B. 材质与外观 (CAT_MATERIAL_LOOK)

- CLEAN_MAINT：清洁/维护成本高
- MATERIAL_RUST：材质/防锈失效
- LOOK_COLOR：外观颜色/做工不满意

C. 履约与服务 (CAT_FULFILL_SERVICE)

- PACK_WEAK : 包装太薄/防护不足
- DOA_DAMAGED : 损坏到货 (DOA/受损)
- SLA_DELAY : 物流时效异常
- CS_POOR : 品牌/店铺服务不达标

D. 适配与场景 (CAT_FIT_SCENE)

- FIT_COMPAT : 尺寸/兼容性不符
- SCENE_ADAPT : 沥水/排水问题

E. 安装与使用体验 (CAT_INSTALL_USE)

- INSTALL_COMPLEX : 安装/组装复杂 (耗时高)
- INSTALL_FIX : 安装/调节/固定性差

F. 信息说明与选型指引 (CAT_INFO_EXPECT)

- PAGE_MISSING : 页面信息/选型指引不清

(以后如果你想继续细分预期类, 也可以把新的预期相关标签挂在这个大类下)

G. 价格与政策 (CAT_PRICE_POLICY)

- RETURN_POLICY_BAD : 退换信息/政策体验不佳
- VALUE_WEAK : 性价比差/不值这个价

5. 整体效果小结

现在这套 v4 体系的特点:

1. 运营端只打“具体问题是什么”

- 不再要求判断“是不是页面写错/图片骗人”, 这些交给你的后续模块。

2. 跟“预期相关”的, 全部通过细分标签来体现

- 比如: 晃 → STABILITY_TIP; 生锈 → MATERIAL_RUST; 觉得不值 → VALUE_WEAK; 信息看不懂 → PAGE_MISSING。

3. 标签名尽量口语化、可视化

- 看名字就有画面感: 晃、重、贵、不值、包装太薄、外观不满意、沥水有问题、页面信息/选型指引不清.....

再次给你新文件链接，方便直接接入系统：

👉 **v4 标签库 JSON:**

[return_dim_tag_v4_mergedInfo_20251117.json](#)

如果你接下来准备上模型 / 规则打标，我也可以基于这 18 个标签帮你写一份：

- 每个标签 **关键词/句式示例**（适合做 prompt 或规则）
- 再配一套 **运营训练题（评论 → 正确标签）**，可以直接拿去做培训。 </>

📄 🍷 🗨️ ⬆️ ↺️ ...

A,D两大类的大类名称是否可以合并到一起？统一叫什么比较好呢？INSTALL_FIX的标签中文名称是否可以更倾向于使用过程调节起来困难？

已思考 38s >

可以合并，而且你这个直觉很对：A+D 本质上是在描述**“这个架子本身长什么样 & 放上去好不好用”**，拆成两个大类对一线来说有点学术了。

我已经按你的想法做了两件事：

1. 合并 A、D 两个大类

原来：

- A: CAT_STRUCT_LOAD 结构与承载
 - CAPACITY_EFF, PARTS_DESIGN, STABILITY_TIP, WEIGHT_MOVE
- D: CAT_FIT_SCENE 适配与场景
 - FIT_COMPAT, SCENE_ADAPT（沥水/排水问题）

✅ **现在统一成一个大类：**

- **category_code:** CAT_STRUCT_FIT
- **category_name_cn:** 「结构与摆放/适配体验」

所有原来在 **结构与承载** 和 **适配与场景** 下的标签，都归到这个大类下面，比如：

- CAPACITY_EFF 容量/有效面积不足（含扩容效果不达标）
- PARTS_DESIGN 分区/配件设计不合理（刀叉篮/杯架等）
- STABILITY_TIP 整体稳定性差/易晃动
- WEIGHT_MOVE 自重过大/移动不便
- FIT_COMPAT 尺寸/兼容性不符
- SCENE_ADAPT 沥水/排水问题

命名逻辑：

- “结构” → 架子本身怎么长、分区/承重/稳定性
- “摆放/适配” → 放在什么水槽/台面/场景下好不好用
- “体验” → 让一线知道这是“使用层面”的问题，而不是售后/价格/页面信息

这样，运营看到这个大类名就能大概知道：

这是关于“架子本体 + 放哪儿 + 怎么放”的问题集合。

2. INSTALL_FIX 中文名称微调，更偏向“使用过程调节不省心”

你提到：

“标签中文名称是否可以更倾向于使用过程调节起来困难？”

在不和 INSTALL_COMPLEX（安装步骤复杂）混淆的前提下，我把 INSTALL_FIX 改成了一个更贴近你意思、又兼顾原定义的说法：

✅ INSTALL_FIX **新中文名称：**

「使用中位置调节/固定不省心」

• 保持的核心含义：

不是“第一次安装很难”，而是：

- 调好了后还老跑位、
- 老要拧螺丝、
- 某些卡扣/支撑调节起来老是锁不牢。
- 与 INSTALL_COMPLEX 的分工会更清晰：
 - INSTALL_COMPLEX：**装起来很折腾**（步骤多、说明难懂、要找工具）
 - INSTALL_FIX：**用着不省心**（经常要重新调/紧固，位置不稳，固定不牢）

原来的 definition / boundary 我没有改逻辑，只是同步更新了 updated_at：

- 仍然是：
“按说明装好之后，位置仍易位移、需频繁紧固或关键部件无法可靠锁定，影响使用稳定性。”

3. 最新 JSON 文件

所有这些修改，我已经写进了新版本的标签库 JSON：

👉 下载（已合并大类 + 修改 INSTALL_FIX 名称）：
[return_dim_tag_v5_structfit_20251117.json](#)

如果你接下来希望，我可以给这 18 个标签做一张「**大类-子标签-一句话解释**」对照表，方便你拿去做运营培训或当系统内的 hover 提示文案。 </>

📄 👍 💬 ↗️ ↺️ ...

就目前的标签体系来说，请你帮我把搭建标签体系的主线逻辑提炼出来，我想在日后其它品类继续复用这个思路

已思考 37s >

我先用一句话总结一下这套体系的“灵魂”：

不是在问“谁的错”，而是在用用户视角，把“具体哪里不好用”拆成若干互斥的小问题，覆盖完整的退货旅程。

你后面换品类，只要抓住这条主线，基本就能复用。

一、三层结构：大框架先定死，再长细节

这套标签体系本质是 **3 层结构**：

1. 顶层：用户旅程维度

- 商品本体：结构/适配、材质/外观
- 安装 & 使用过程

- 信息说明 & 选型指引
- 履约 & 服务
- 价格 & 退换政策

2. 中层：问题大类 (category_code)

当前大类是：

- CAT_STRUCT_FIT：**结构与摆放/适配体验**
- CAT_MATERIAL_LOOK：**材质与外观**
- CAT_INSTALL_USE：**安装与使用体验**
- CAT_INFO_EXPECT：**信息说明与选型指引**
- CAT_FULFILL_SERVICE：**履约与服务**
- CAT_PRICE_POLICY：**价格与政策**

3. 底层：具体问题标签 (tag_code)

每个 tag 都是用户能说出来的一句抱怨，如：

- “容量太小” → CAPACITY_EFF
- “架子晃/不稳” → STABILITY_TIP
- “调好总跑位” → INSTALL_FIX（现在叫：**使用中位置调节/固定不省心**）
- “装起来太折腾” → INSTALL_COMPLEX
- “外观颜色/做工不满意” → LOOK_COLOR
- “页面信息/选型指引不清” → PAGE_MISSING
- ...

主线逻辑就是：

先把用户从“看到页面 → 收货 → 使用 → 售后/退货”的路径拆成几个「阶段维度」，再在每个维度下，用用户语言定义「一个个可观察的问题点」。

二、核心设计原则：以后做新类目也沿着这几条

1. 只打“体验问题”，不打“页面对不对”

我们刻意把这两块拆开了：

- ****打标体系：只负责记录**
 - 到手的東西是什么样
 - 用起来哪里不顺
 - 服务/物流/价格体验如何
- **页面描述是否准确：**

- 交给你后面单独的“页面描述匹配评估”去做。

所以我把“规格参数与宣传不一致 / 图片不符”这类标签从人工体系里拿掉，避免运营在那边纠结：

“到底是他预期太高，还是页面真写错了？”

之后所有“描述不符”由算法/规则判定；

人工标签只关注“事实上的体验问题”。

2. 区分“过程痛苦”和“结果不稳”

你在水槽架这个品类里，其实有两种典型抱怨：

- “装起来心很累” → INSTALL_COMPLEX
- “装完了还老要拧螺丝/总跑位” → INSTALL_FIX
- “放上去就晃/一边悬空” → STABILITY_TIP / FIT_COMPAT

我们给每种「不同性质的痛点」一个独立标签，并在边界里写成**决策树**：

- 装之前就痛苦 → 安装复杂
- 装完还要折腾 → 位置调节/固定不省心
- 自身结构软、晃 → 整体稳定性差
- 根本放不下/太大太小 → 尺寸/兼容性不符

以后换品类（比如小家电）：

同样可以沿着这条思路，把「装/调/用」拆开，而不是一个大“安装问题”糊掉。

3. 一切标签都围着“用户话术”命名

你可以看到，我们调整后的中文名都尽量是一句“吐槽语句的压缩版”，例如：

- **结构与摆放/适配体验**（大类）
 - 容量/有效面积不足
 - 分区/配件设计不合理
 - 整体稳定性差/易晃动
 - 自重过大/移动不便
 - 尺寸/兼容性不符
 - 沥水/排水问题
- **安装与使用体验**
 - 安装/组装复杂（耗时高）
 - 使用中位置调节/固定不省心

- **材质与外观**
 - 材质/防锈失效
 - 外观颜色/做工不满意
 - 清洁/维护成本高
- **信息说明与选型指引**
 - 页面信息/选型指引不清
- **履约与服务**
 - 包装太薄/防护不足
 - 损坏到货 (DOA/受损)
 - 物流时效异常
 - 品牌/店铺服务不达标
- **价格与政策**
 - 性价比差/不值这个价
 - 退换信息/政策体验不佳

以后做新类目，只要问自己一句话：

“这个标签名字能不能直接复读成用户的一句抱怨？”

如果不能，就再口语化一点。

4. MECE 思路：每个问题只属于一个“最小格子”

这套体系里有几个关键拆分，你可以在别的品类照抄：

- **稳定 vs 容量 vs 适配 vs 分区设计：**
 - 放不稳、会晃 → 稳定
 - 放得下但太小/浪费面积 → 容量
 - 根本放不下、太大太小 → 适配
 - 明明空间够，但分区/挂件几何奇怪 → 分区设计
- **安装过程 vs 使用过程：**
 - 安装步骤多、说明看不懂 → 安装复杂
 - 装好后老要调 → 位置调节/固定不省心
- **包装 vs DOA：**
 - 包装看着很危险/纸箱软/缓冲少，但东西还能用 → 包装太薄/防护不足
 - 东西本体真坏了/少件 → 损坏到货 (DOA)
- **价格 vs 预期 vs 具体问题：**
 - 贵/不值/不划算 → 性价比差

- 哪里不好用 → 具体功能标签（容量/稳定/材质/清洁难...）
- 预期相关不单做一个标签，而是靠**功能标签 + 星级 + 文字**组合，在分析层面感知。

这样就避免一个抱怨被打到三个维度里，后面分析会很干净。

三、复用到其他品类的步骤建议

以后你换成别的品类（例如：锅具、小家电、浴室收纳、办公椅...），可以按这个步骤来：

步骤 1：沿用顶层大类不动

直接继承这 6 个维度作为「骨架」：

1. 结构与摆放/适配体验（结构 & 功能表现 & 场景适配）
2. 材质与外观
3. 安装与使用体验
4. 信息说明与选型指引
5. 履约与服务
6. 价格与政策

不管什么品类，用户退货原因 90% 都能归进这六格。

步骤 2：对新类目，先用这 6 格聚类真实退货原因

拿一批这个类目的退货评论，做一个简单的“人工聚类”：

- 属于「本体结构/功能问题」的 → 丢进 结构与摆放/适配体验
- 属于「外观/材质/做工」的 → 丢到 材质与外观
- 属于「安装/使用过程」的 → 丢到 安装与使用体验
- ... 依此类推。

然后看看每个大类里，反复出现的、可清晰命名的痛点有哪些，就形成新的 tag_code。

步骤 3：能复用的直接复用，只在必要时新增

- 类似“性价比差”“物流慢”“客服差”“包装太薄”等是 **跨品类通用** 的，直接沿用。
- “外观颜色/做工不满意”“清洁难”也是高度通用，可以原样复用。
- 对于每个品类特有的问题，只在**频次够高**时再加新标签，比如：
 - 电器类可能需要新增“噪音大”“异味严重”“发热严重”等；
 - 家具类可能需要“坐感不舒服”“支撑不够/塌陷”等。

步骤 4：为每个新标签写清楚“边界三句话”

每个新标签都写一个 mini 决策树：

- 1. 这个标签“是什么”（1 句话定义）
- 2. “不是它而是另一个”的例子（对容易混淆的标签写清这里不归我管）
- 3. 典型用户话术示例（1-2 个例句）

运营培训的时候，就拿这三句话说清即可。

四、总结一句话的主线逻辑

先用用户旅程定“几个大抽屉”，再用用户吐槽语定义“每个抽屉里的小格子”。
小格子永远描述“事实上的体验问题”，
页面对不对、预期高不高交给别的模块处理。

照着这条主线走，你在别的品类里，只需要替换“小格子”，大框架几乎不用重造。



↳ **主线逻辑就是：先把用户从“看到页面 → 收货 → 使用 → 售后/退货”的路径拆成几个「阶段维度」，再在每个维度下，用用户语言定义「一个个可观察的问题点」。

按照这个主线逻辑，目前这套标签体系的映射关系是？

已思考 32s >

好，我们就按你说的这条主线——

「看到页面 → 收货 → 使用 → 售后/退货」

来把现在这套标签体系的映射关系梳理一遍。

0. 先回顾一下当前的大类 & 标签（v5 版）

大类（category_code → category_name_cn）

- CAT_STRUCT_FIT：结构与摆放/适配体验
- CAT_MATERIAL_LOOK：材质与外观
- CAT_INSTALL_USE：安装与使用体验

- CAT_INFO_EXPECT: 信息说明与选型指引
 - CAT_FULFILL_SERVICE: 履约与服务
 - CAT_PRICE_POLICY: 价格与政策
-

1. 阶段一：看到页面（决策前）

用户通过页面信息 & 选型做购买决策

对应大类

- CAT_INFO_EXPECT: 信息说明与选型指引

对应标签

- PAGE_MISSING — 页面信息/选型指引不清
 - 页面在尺寸、适配范围、安装要点（是否要自己装）、多尺寸/多版本差异等信息上说得不清楚 / 看不懂，
 - 导致用户选错型号 / 误判是否适配 / 误判安装难度。

这个阶段对应的是“我看了页面再下单”这一步里，信息是否足够清晰。

真正的“描述不符（规格写错/图片欺骗）”已经从标签里剥离出去，交给后续页面匹配模块。

2. 阶段二：收货（物流 & 包装 & 到货状态）

从发货 → 运输 → 开箱这一段体验

对应大类

- CAT_FULFILL_SERVICE: 履约与服务

对应标签

- SLA_DELAY — 物流时效异常
 - 到货晚、卡单、跟承诺时效差距大。
- PACK_WEAK — 包装太薄/防护不足
 - 箱子很薄、内衬少、看着就容易磕碰；
 - 或已经造成轻微外观损伤但功能还可以用。
- DOA_DAMAGED — 损坏到货（DOA/受损）

- 开箱就发现**缺件/变形/坏了/无法正常安装使用**。
- 包括本应出厂就装好的关键配件（脚垫、连接件）缺失/装错，导致一开箱就不正常。

这部分是典型的“收货体验”维度，跟产品设计本身分开看。

3. 阶段三：安装 & 使用过程（核心体验）

我把它拆成两个层次：

3.1 产品本体：结构 & 适配 & 使用表现

「东西本身长得怎么样、放哪儿、好不好用」

大类：CAT_STRUCT_FIT — 结构与摆放/适配体验

- **CAPACITY_EFF — 容量/有效面积不足（含扩容效果不达标）**
 - 能放的东西比想象中少、空间浪费严重、扩展后有效面积依然不够用。
- **PARTS_DESIGN — 分区/配件设计不合理（刀叉篮/杯架等）**
 - 刀叉篮/杯架高度、宽度、开孔设计不合理，导致**放不稳/难拿/压缩空间**。
- **STABILITY_TIP — 整体稳定性差/易晃动**
 - 架子整体发软、晃、易倒、一边悬空等。
- **WEIGHT_MOVE — 自重过大/移动不便**
 - 太重、抓不住、从一个位置挪到另一个位置很费劲。
- **FIT_COMPAT — 尺寸/兼容性不符**
 - 跟水槽/台面尺寸不配套：**放不下/太小/缝隙太大严重影响使用**。
- **SCENE_ADAPT — 沥水/排水问题**
 - 水不往水槽里流，滴到台面/地面，容易积水、干得慢等**水路问题**。

3.2 安装 & 使用过程中的操作负担

「装的过程 / 用的过程中调节是不是很折腾」

大类：CAT_INSTALL_USE — 安装与使用体验

- **INSTALL_COMPLEX — 安装/组装复杂（耗时高）**
 - 步骤多、说明难懂、要找各种工具，用户典型感受是：“装起来太折腾 / 费劲 / 费时间”。

- **INSTALL_FIX — 使用中位置调节/固定不省心**

- 好不容易装好了，结果：
 - 架子总移位
 - 螺丝/卡扣总要重新拧
 - 高度/位置调节不顺滑/锁不牢

这里的主线逻辑是：

- “过程痛苦 → INSTALL_COMPLEX”
- “结果不省心（老要调） → INSTALL_FIX”
- “自身结构不稳 → STABILITY_TIP”
- “尺寸根本不合 → FIT_COMPAT”

3.3 材质 & 外观 & 清洁

「看上去怎么样、摸起来/洗起来怎么样」

大类：CAT_MATERIAL_LOOK — 材质与外观

- **MATERIAL_RUST — 材质/防锈失效**
 - 用没多久就生锈、掉皮、材质明显不耐用。
- **LOOK_COLOR — 外观颜色/做工不满意**
 - 颜色不好看、显廉价、焊点粗糙、喷涂不均等主观嫌弃长相/做工的反馈。
- **CLEAN_MAINT — 清洁/维护成本高**
 - 缝隙多、积水/积垢难清理，擦洗成本高。

这些都是“实际拿在手上 & 日常使用时”才会暴露的体验。

4. 阶段四：售后/退货 & 价格感受

4.1 履约 & 售后体验

大类：CAT_FULFILL_SERVICE — 履约与服务

- **CS_POOR — 品牌/店铺服务不达标**
 - 客服慢、态度差、推诿、不按承诺处理等。

收货时的 DOA/包装/时效也在这个大类里（上面第二段已经列出）。

4.2 价格与退换政策

大类：CAT_PRICE_POLICY — 价格与政策

- **VALUE_WEAK — 性价比差/不值这个价**
 - 用户明确表达“贵/不值/不划算/不会再买”。
 - **RETURN_POLICY_BAD — 退换信息/政策体验不佳**
 - 退换流程复杂、必须去很远的网点、只支持某快递点、周期长等。
- 这一段就是用户在「考虑退不退 / 怎么退」时，对**钱 & 规则**的真实感受。

5. 映射关系总结成一张“路径 → 大类 → 标签”骨架

你以后做其它品类，可以直接照着这个骨架：

1. 看到页面（决策前）

- 信息说明与选型指引（CAT_INFO_EXPECT）
 - 页面信息/选型指引不清（PAGE_MISSING）

2. 收货阶段（物流 & 包装 & 到货状态）

- 履约与服务（CAT_FULFILL_SERVICE）
 - 物流时效异常（SLA_DELAY）
 - 包装太薄/防护不足（PACK_WEAK）
 - 损坏到货（DOA_DAMAGED）

3. 安装 & 使用阶段（核心体验）

- 结构与摆放/适配体验（CAT_STRUCT_FIT）
 - 容量/有效面积不足（CAPACITY_EFF）
 - 分区/配件设计不合理（PARTS_DESIGN）
 - 整体稳定性差/易晃动（STABILITY_TIP）
 - 自重过大/移动不便（WEIGHT_MOVE）
 - 尺寸/兼容性不符（FIT_COMPAT）
 - 沥水/排水问题（SCENE_ADAPT）
- 安装与使用操作体验（CAT_INSTALL_USE）
 - 安装/组装复杂（INSTALL_COMPLEX）
 - 使用中位置调节/固定不省心（INSTALL_FIX）
- 材质与外观（CAT_MATERIAL_LOOK）
 - 材质/防锈失效（MATERIAL_RUST）
 - 外观颜色/做工不满意（LOOK_COLOR）
 - 清洁/维护成本高（CLEAN_MAINT）

4. 售后/退货 & 价格阶段

- 履约与服务 (CAT_FULFILL_SERVICE)
 - 品牌/店铺服务不达标 (CS_POOR)
- 价格与政策 (CAT_PRICE_POLICY)
 - 性价比差/不值这个价 (VALUE_WEAK)
 - 退换信息/政策体验不佳 (RETURN_POLICY_BAD)

你以后在别的品类，只要：

1. 先确定这四个阶段还成立（基本都会成立），
2. 再看每个阶段下 **哪些标签可沿用、哪些要新增**，

这套主线逻辑就能“像模具一样”复用下去。需要的话，我也可以帮你按这个骨架，画一张可放进 PPT 的「标签体系概览图」。



沥水/排水问题，的tag code SCENE_ADAPT是否需要修改一下，这个不太贴合

已思考 52s >

我分两层跟你说：“要不要改”和“如果改，怎么改最稳妥”。

1. 要不要改 SCENE_ADAPT 这个 tag_code?

现在这个标签的中文名和定义已经收窄成：

- **中文名：**沥水/排水问题
- **含义：**水往哪儿流、流得顺不顺（滴台面、积水、流不到水槽）

SCENE_ADAPT 这个 code 对应的是“场景适配”，确实已经**和现在语义不贴了**。

从两个角度看：

1. 工程 &数据稳定角度：

- tag_code 一般当作“主键 / ID”，改名会涉及：

- 历史数据迁移
- 报表、埋点、模型特征等的映射
- 所以**不建议“直接改 code”**，除非你确定相关链路都能一起改。

2. 长期语义 & 可复用角度：

- 你未来打算这套体系**跨品类复用**，tag_code 如果语义歪了，会让后续人接手很痛苦。
- “沥水/排水问题”跟 SCENE_ADAPT 的词根已经没什么关系了，从**规范化视角**确实值得修一修。

我的建议是：

不要在原记录上硬改 code，
而是走“新增一个语义干净的 code + 废弃老 code”的路径。

2. 建议的新 tag_code 以及迁移方式

2.1 推荐的新 tag_code

参考你现有命名风格（CAPACITY_EFF, MATERIAL_RUST, PACK_WEAK 这种“名词+弱点”的结构），我会推荐：

- DRAIN_FLOW
 - 英文含义：Drainage flow（排水流向/排水表现）
 - 对应中文名：**沥水/排水问题**（保持不变）

如果你更想强调“外漏”，可以选 DRAIN_LEAK，
但考虑到你这个标签还管“积水不干”，DRAIN_FLOW **覆盖面更大**。

2.2 迁移建议（最稳的一条路）

1. 新增一条标签：

- tag_code：DRAIN_FLOW
- tag_name_cn：沥水/排水问题
- category_code：仍然挂在 CAT_STRUCT_FIT（结构与摆放/适配体验）
- definition & boundary_note：直接沿用现在 SCENE_ADAPT 的文案

2. 将 SCENE_ADAPT 标记为历史标签：

- 保留它在表里，但：
 - is_active：设为 0

- `effective_to`: 填一个截止日期
- `definition` 里可以补一句:

“已由 `DRAIN_FLOW` (沥水/排水问题) 替代, 仅用于历史数据兼容。”

3. 数据/系统侧的做法:

- 历史数据里:
 - 可以在报表层做一个简单映射: `SCENE_ADAPT` → `DRAIN_FLOW`
- 新的数据流: