

智能动态网格系统

一、策略核心理念

1.1 策略本质：结构驱动的自适应网格系统

本策略是一个结合结构、波动、仓位加权和动态重置的 **区间—趋势双模态量化交易系统**。

通过高时间框架（4h/1d/15m）识别关键价格结构（支撑/阻力），利用 1m 执行精细化网格交易，实现：

- 区间（RANGE 模式）：稳定套利 + 高频均值回归
- 趋势（DGT 模式）：自动切换结构、跟随趋势

策略核心哲学：

“在结构内吃波动，在结构外跟趋势。”

1.2 多周期结构（S/R）为核心锚点

系统使用高时间框架的 pivot(20,20) 结构识别：

- 支撑
- 阻力
- 区间有效性
- 中轴（midline）

并通过 **20 bar 延迟确认机制** 彻底避免未来函数。

结构在整个策略中扮演“地形地图”的角色：

- 网格区间由结构定义
- 仓位权重由结构形状决定
- DGT 重置时重新生成结构性区间
- 风险由结构有效性约束

1.3 网格执行层：几何价格 + 结构加权仓位

网格的价格与仓位设计遵循三大原则：

① 几何价格间距

以当前价格 P0 生成双向几何序列：

- 价格越远间距越大（符合真实市场的波动结构）
- 单边最多 10 层（可配置）
- 仅保留落在 S/R 区间内的网格

几何网格优于线性网格，因为：

- 远端挂单更稀疏
- 波动越大越能捕捉趋势性摆动
- 低端拥挤、高端发散 → 更符合涨跌分布

② 边缘加权仓位 (Edge Weighting)

在区间边缘投入更多仓位更有优势：

- 靠近支撑 → 多头权重最大
- 靠近阻力 → 空头权重最大
- 中轴附近权重最小（避免过度交易）

仓位乘以权重后归一化：

仓位不是均等分布，是结构驱动的非对称漏斗分布

③ 命中衰减 (Hit Decay)

一个网格层被反复成交，代表：

- 局部噪声偏多
- 波动在缩窄
- 套利空间变小

因此系统自动衰减该价格层权重：

```
w_decayed = w_raw * exp(-hits / decay_k)
```

衰减机制的作用：

- 防止单价格层反复建仓（库存累积风险）
- 自动减少高频震荡中的不必要交易
- 越常被触发 → 仓位越轻

1.4 RANGE 模式：区间内吃波动

当价格在支撑与阻力之间：

- 构造几何网格

- 结构加权仓位
- 衰减历史命中
- 保持 inventory 平衡
- 执行 Maker 网格套利

RANGE 模式目标：稳定吃区间内震荡的双边价差

1.5 DGT 模式：突破后动态重置网格（Dynamic Grid Tilt）

DGT 是本策略的关键创新：

当价格突破结构区间后，自动将网格“迁移”到新的价格中心，并按趋势方向倾斜仓位结构。

DGT 的三个核心思想：

① 新中心价格（New Grid Center）

突破后不再使用旧结构，而是：

- 用突破后的价格
- 或最近 swing high/low
- 或突破后的成交均价
作为新网格中心。

② 新区间动态生成（Dynamic Range Rebuild）

突破后自动生成新的网格区间：

- 上涨突破 → 网格上移至新价格区间
- 下跌突破 → 网格下移至下一区间

区间由以下决定：

- ATR × 系数
- PCA 波动结构
- 或按一定 % 布局（如 ±2% 区间）

③ 趋势方向的倾斜仓位结构（Tilt）

在趋势方向：

- 增加挂单密度
- 增加仓位权重

与原始 RANGE 模式倾向相反：

- RANGE：两边权重高，中轴低

- DGT：顺势一边权重高，逆势一边权重低

这样你能：

- 顺势吃趋势的 pullback
- 不被区间边界卡仓
- 也避免方向性过度暴露

1.6 区间 + 趋势双模态的优势

传统网格的缺点：

- 趋势行情被吃掉（越跌越买、越涨越卖）
- 越走越亏
- 死在趋势里

本策略通过：

- 结构识别（固定区间）
- DGT（趋势漂移）
- 假突破过滤（避免骗你跳结构）
- 仓位上限（避免累积方向性风险）

实现：

| 区间稳定盈利 + 趋势不爆仓 + 趋势中反而赚钱。

二、策略整体架构（Strategy Architecture）

本策略由 **结构识别层（Structure Layer）** → **执行层（Execution Layer）** → **动态重置层（DGT Layer）** → **风控层（Risk Layer）** → **仓位管理层（Position Layer）** 五大核心模块构成。

其整体运作逻辑如下：

2.1 结构识别层（Structure Detection Layer）

主要功能：

- 识别关键 S/R（支撑/阻力）
- 定义区间边界
- 提供 RANGE 模式基础框架
- 作为 DGT 的结构对齐依据

核心特征：

使用高时间框架（默认：4h，可切换1d或15m）的pivot高低点，通过以下机制识别结构：

1. pivot(20,20)

- 左右各20根K线
- 用close得出pivot_high / pivot_low

2. +20 bar确认 (avoid future leak)

所有结构必须“延迟20根”才生效。

3. merge_asof下采样

把高周期结构广播到低周期执行（1m）。

2.2 执行层（Execution Layer）

主要功能：

- 在区间内生成几何网格
- 按照权重分配仓位
- 刷新挂单、执行成交
- 控制inventory分布

执行流程：

763. 每1m bar刷新结构值（support、resistance、midline）

764. 判断区间是否有效

765. 根据几何网格生成所有买卖层价格

766. 分配结构加权 + 衰减后仓位

767. 下达挂单（limit maker）

768. 回调成交，记录权重衰减

769. 实时计算净仓位、风险暴露

2.3 动态重置层（DGT Layer）（本策略的关键创新）

主要功能：

- 当价格突破结构时，“删除旧区间，构建新区间”
- 实现趋势行情的自适应网格中心移动

- 避免传统网格在趋势行情中爆仓

流程要点：

- 检测突破 (≥ 2 根 + ATR 扩张 + 量价确认)
- 触发 DGT 模式
- 新网格中心 = 突破后的平衡价格
- 新区间 = $\pm (\text{ATR} \times \text{系数})$ 或 $\pm (\text{固定百分比})$
- 新网格按照趋势方向倾斜 (Tilt)
- 顺势方向权重更高
- 逆势方向几乎不挂

DGT 是“趋势中的网格”，让你：

| 在上涨趋势中做 pullback 买入，在下跌趋势中做反弹卖出。

2.4 风控层 (Risk Layer)

风控层是一个贯穿所有流程的“实时过滤器”，包括：

(1) 突破检测 (Breakout Detect)

价格是否越界？越界是否有效？是否假突破？

(2) 假突破过滤 (三重过滤)

- 价格
- 波动
- 成交量

全部满足才算突破有效并进入 DGT。

(3) 微震荡冷却 (Micro Oscillation Cooldown)

当短期波动不足以支撑网格交易时 (ATR 收缩) 暂停下单。

(4) 中轴去库存 (Midline Inventory Reset)

防止价格回到中轴时仍持有大额偏向仓位。

(5) 方向性风险上限 (Directional Exposure Limit)

限制 net_position 最大值，避免趋势反转时爆仓。

2.5 仓位管理层 (Position Management Layer)

负责最终决定每一层网格挂多少单。

由以下三部分组成：

(1) 边缘加权 (Edge Weighting)

公式：

代码块

```
1 w_long(p) = ((resistance - p) / (resistance - support))α
2 w_short(p) = ((p - support) / (resistance - support))α
```

本质是：

越靠近边界挂越多，越靠近中轴挂越少

符合量价结构和真实市场行为。

(2) 命中衰减 (Hit Decay)

多次成交的价格层：

- 命中次数 ↑
- 衰减指数 ↓
- 仓位权重自动变小

表达式：

代码块

```
1 w_decayed = w_raw * exp(-hits / decay_k)
```

这是你策略最强的“反 over-trading”机制。

(3) 仓位归一化 + 风险上限

所有权重通过：

代码块

```
1 normalized_w = w_decayed / Σ(w_decayed)
```

然后乘以：

- 总仓位预算
- 单格最大仓位
- 最大方向性仓位上限
- 中轴去库存约束

最终确定每一层挂单数量。

2.6 策略运行状态机 (State Machine)

策略整个流程由 4 个状态组成：

① RANGE 状态 (区间网格)

- 支撑/阻力有效
- 价格在区间内
- 执行结构加权网格

② BREAKOUT 突破状态 (确认突破中)

- 进入突破过滤
- 观察 1~2 根确认 K
- 如果突破成立 → 切 DGT
- 如果是假突破 → 回到 RANGE

③ DGT 状态 (动态趋势网格)

- 新中心
- 新区间
- 顺势网格
- 仓位倾斜

当趋势结束或进入新结构 → 回到 RANGE。

④ COOL 冷却状态 (低波动)

- ATR_ratio < 阈值
- 暂停挂新单，但保留旧订单
- 当波动恢复 → 回到 RANGE/DGT

三、动态结构网格 (DGT) 机制

本章介绍结构自适应动态网格策略（Structure-Adaptive Dynamic Grid Strategy）中的核心创新——**动态网格重置（Dynamic Grid Reset）机制，即 DGT 思想**。这一机制解决了传统网格策略最根本的风险点：**区间结束 = 策略死亡**，并让网格在趋势中保持 *alive* 状态，使策略在趋势行情中仍能持续捕捉套利机会，而不是被动终止。

3.1 传统网格策略的问题

传统网格策略有两个致命弱点：

(1) 区间突破意味着策略终止（不再盈利）

传统策略在价格突破上沿或下沿时，会：

- 上破 → 卖空所有持仓 → 策略死亡
 - 下破 → 用光现金买满币 → 策略死亡
- 从此以后没有新的套利机会
→ 策略收益停止增长

这正是论文《Dynamic Grid Trading Strategy》证明传统网格**期望值为零**的重要原因

(2) 网格中心是静态的，不随市场移动

传统网格假设价格在区间内上下震荡，但真实世界：

- 趋势行情多于震荡行情
- 当中心价格偏移后，网格变得无效
- 静态中心 → 买到越来越多的“垃圾仓位”
- 仓位积累风险不断上升

3.2 DGT：动态中心与区间重置（核心思想）

DGT (Dynamic Grid-based Trading) 提出：

只要区间被突破，就立即重置网格，把突破点作为新中心点。

其核心逻辑如下：

(1) 价格上破区间 → 重置中心到当前价

传统网格上破会清仓结束。但 DGT 认为：

- 上破 = 市场进入更高均值区间
- 不清仓，而是 **把利润沉淀并重新部署新网格**

动作：

- 保留套利赚到的钱 (wallet)
- 初始资本恢复 (input-money 被复位)
- 当前突破价成为新中心
- 重新打开一个新的几何网格

效果：

- 趋势行情中不断滚动盈利，不会策略死亡
- 累积小赚 → 变成长期大赚

这正是论文中 DGT 能持续跑赢行情的关键

(2) 价格下破区间 → 币持仓保留、套利利润成为新本金

传统网格下破会买满币并停止。

DGT 的做法：

- 原网格中积累的利润 wallet 保留
- 因为跌破区间，持仓会越来越多 → 允许
- 但 不强制卖出，也不结束策略
- 用 “套利利润” 作为下一个网格的新本金

逻辑：

- 下跌时本来就应该 “买得更多”
- DGT 遵循这一思想，不背离网格本质（买跌卖涨）

因此在熊市也可以：

- 降低持仓均价
- 维持套利效率
- 提高回本速度
- 等待下个震荡或反弹阶段继续盈利

3.3 为什么 DGT 能够提升网格策略的期望值？（理论解释）

论文证明传统网格期望值 = 0，因为：

一旦边界突破，套利次数有限 → 策略停止。

而 DGT 的核心突破点：DGT 把 “有限套利次数” 变成 “无限套利次数”。

如何做到？

- 每次突破后立即重建新网格

- 继续捕捉新的套利机会
- 无限循环

数学上：

网格的期望值 = 套利次数 \times 单次收益

传统网格套利次数有限 $\rightarrow EV = 0$

DGT 套利次数趋近无限 $\rightarrow EV > 0$

(引用论文中 2.2.1、2.2.2 的期望证明)

因此，DGT 能长期跑赢市场，是数学推导出来的必然结果，而不是偶然。

3.4 DGT 如何与“结构自适应网格”结合？

DGT 只负责：

- 价格突破 \rightarrow 重置
- 新中心价 = 突破价
- 持续创造套利次数

但在你的策略中，我们加入更强逻辑：

(1) 不是所有突破都应重置 \rightarrow 我们用结构 (S/R) 判断真假突破

- 如果突破来自结构支撑/阻力 \rightarrow 多半为趋势启动 \rightarrow 应重置
- 如果突破来自区间中轴附近 \rightarrow 多半是假突破 \rightarrow 不应重置

(2) 结构位置本身也会影响网格间距与仓位权重

- 越靠近支撑，网格越密集、权重越高
- 越靠近阻力，网格越轻仓、风险越低

这使得我们的策略比论文的 DGT 更高级：

不是盲目重置，而是结构驱动的自适应重置。

3.5 DGT 在策略中的整体流程

2191. 价格接近 S/R \rightarrow 开启网格结构判断

2192. 若突破结构 \rightarrow 启动 DGT 重置机制

2193. 突破点成为新的中心价

2194. 根据新结构重新生成 S/R 区间

2195. 网格自动调整（间距、仓位权重、风控）

2196. 继续套利循环

2197. 重复直到结构破坏 → 再重置

等价于：

→ 结构提供“方向框架”

→ DGT 提供“无限循环能力”

→ 网格提供“套利引擎”

这三者结合，构成你现在的：**结构自适应动态网格策略**

四、网格生成与仓位分配

本章详细定义网格的两大组成部分：

235. 网格价格 (Grid Prices)

236. 网格仓位 (Grid Weights & Allocation)

并说明如何在 DGT 效应下进行自适应重建 (Dynamic Reset)。

4.1 网格价格生成 (Grid Price Construction)

网格使用**几何间距 (Geometric Spacing) **而非线性间距。

4.1.1 为什么使用几何网格？

几何网格更接近真实市场特征：

- 价格越远 → 波动越大 → 挂单自然应该更稀疏
- 越接近当前价 → 频率越高 → 挂单更密集
- 可避免线性网格在远端堆积太多无效单
- 在趋势行情下更适合动态移动 (DGT)

数学上，几何间距比线性间距具有更高的回本概率和更稳定的 inventory 行为。

4.1.2 网格价格公式

以当前价格 P_0 为中心构造：

向上网格：

```
P_sell[k] = P0 × (1 + g)^k
```

向下网格：

```
P_buy[k] = P0 × (1 - g)^k
```

其中：

- $g = \text{grid_gap_pct}$ (默认: $0.18\% = 0.0018$)
- k = 网格层级编号 ($0, 1, 2, \dots$)

优点：

- 增长或下降是指数的 → 远端网格自然变宽
- 简洁、可控且易于加入 DGT 偏移

4.2 网格截断 (Grid Truncation via Structure)

生成的网格需要根据结构区间进行筛选。

网格仅在：

代码块

```
1 support <= price <= resistance
```

范围内生效。

意义：

- 保证网格交易发生在“结构有效区域”
- 避免在趋势极端区域挂不必要的深度单
- 与 DGT 的重置机制自然对齐：
DGT 之后区间重新生成 → 网格重新筛选

4.3 网格层级上限 (Max Levels per Side)

为了避免深陷低效长尾区域：

```
max_levels_side = 10
```

单侧最多 10 层。

此参数也与 DGT 配合：

- 趋势中网格会被自动推移

- 不需要大量远端挂单
- 资源占用更低，逻辑更干净

4.4 网格仓位 (Grid Position Allocation)

仓位分配是策略收益与风险的核心部分。

本策略的仓位分配包括三个组件：

1365. 边缘加权 (Edge Weighting)

1366. 命中衰减 (Hit Decay)

1367. 统一归一化 (Normalization)

三者结合形成 动态非均衡 (Non-Uniform) 网格仓位结构。

4.5 边缘加权 (Edge Weighting)

支撑 (support) 与阻力 (resistance) 是结构的两端。

原则：

✓ 越靠近边缘 → 越应挂更大仓位

✓ 越靠近中轴 → 越应减少仓位

因为：

- 边缘处反转概率更高
- 中轴处噪声更多、胜率更低
- 真实人类交易者的行为就是在边缘更积极，在中轴更被动

仓位权重计算如下：

多单权重：

代码块

```
1 w_long(p) = ((resistance - p) / (resistance - support))^α
```

空单权重：

代码块

```
1 w_short(p) = ((p - support) / (resistance - support))^α
```

其中：

- α 为权重指数（默认：2.0）
- 越大 \rightarrow 越强调区间两端
- 越小 \rightarrow 越平缓接近均衡分布

4.6 命中衰减 (Hit Decay)

每当一个网格单被成交，该价格层的命中次数 (hits) +1。

下一次计算时：

代码块

```
1 w_decayed = w_raw * exp(-hits / decay_k)
```

关键意义：

- 越频繁被打到的格子，仓位越小
- 自然抑制噪声震荡中的无效交易
- 自动调节交易密度 \rightarrow 越震荡越轻仓
- 越接近趋势启动 \rightarrow 权重恢复增长

这使你的策略具有智能再平衡能力。

4.7 仓位归一化 (Normalization)

所有网格层最终通过：

代码块

```
1 normalized_w = w_decayed / Σ(w_decayed)
```

计算出每一层的最终仓位比例。

最终实际数量：

代码块

```
1 qty[k] = order_size * normalized_w[k]
```

4.8 仓位分配的安全措施 (Risk-Adjusted Weighting)

为了配合 DGT 与趋势行情稳定运行，本策略加入三大安全机制：

(1) 方向性仓位上限 (Directional Exposure Limit)

代码块

```
1 |net_position| <= exposure_limit * max_position
```

避免在趋势中仓位累积过重。

(2) 边缘仓位上限 (Edge Cap)

即使边缘权重很大，也不能超过设定上限：

代码块

```
1 w_edge <= edge_cap_ratio
```

(例如 30%)

避免结构失效时的极端亏损。

(3) 中轴去库存 (Midline Inventory Reset)

如果价格接近 midline 且净仓位过大 → 自动部分平仓。

防止仓位一直堆在边缘位置。

4.9 DGT 重置后的网格再构造 (关键！)

当突破成立且进入 DGT 模式后：

(1) 新的中心价格 (New Center)

默认使用：

代码块

```
1 new_center = current_price
```

或可配置：

- 突破后的均价 (VWAP)
- 突破前 swing high/low

- 结构突破幅度 × 系数

(2) 新的区间范围 (New Range)

两种方式：

A. ATR 动态区间

代码块

```
1 new_support = new_center - ATR × r
2 new_resistance = new_center + ATR × r
```

B. 百分比区间

代码块

```
1 new_support = new_center × (1 - r)
2 new_resistance = new_center × (1 + r)
```

其中 $r=1.5\% \sim 3\%$ 可选。

(3) 新的网格仓位分布 (Tilted Weights)

顺趋势方向加重：

- 上涨趋势 → 多单侧权重更大
- 下跌趋势 → 空单侧权重更大

权重表达式加入趋势倾斜因子 τ ：

代码块

```
1 w_long_final = w_long × τ_up
2 w_short_final = w_short × τ_down
```

其中：

- 上涨趋势： $\tau_{up} > 1, \tau_{down} < 1$
- 下跌趋势：反之
- 震荡： $\tau \approx 1$, 不启用倾斜

(4) 衰减值是否清零? (Decay Reset)

对于高级版本，我们建议：

- **价格突破后 → 清空衰减记录 (hit_count)**

因为旧区间的噪声与新趋势无关。

但如果策略倾向 conservative 可保留部分衰减（20~30%）。

五、订单管理机制 (Order Management Layer)

订单管理层负责网格策略中所有挂单、取消和成交相关的执行逻辑，是本策略的“执行引擎 (Execution Engine)”。

本章将定义：

- **挂单的生命周期 (Lifecycle)**
- **网格刷新与订单替换机制**
- **成交处理 (Trade Fill Handling)**
- **镜像补单 (Symmetric Replenishment)**
- **DGT 状态下与 RANGE 状态下的不同挂单行为**
- **防滑点机制 (Slippage Protection)**

该机制必须在 1m 级别上高效运行，并确保策略在低波动与高波动环境下均保持一致性。

5.1 订单类型 (Order Types)

本策略仅使用一种订单类型：限价单 (Limit Order) —— Maker 优化

理由：

- 与网格逻辑完美匹配
- 交易成本最低 (Maker 费率通常 0.02% 或更低)
- 可以精准控制成交价，不会因瞬时波动吃到劣势价
- 可以确保成交后不会出现滑点

因此 所有订单均使用 Limit-Maker。No Market Orders.

5.2 挂单生命周期 (Order Lifecycle)

所有订单在每根 1m K 线触发一次生命周期更新。

流程如下：

代码块

1 下一根 1m → 自动取消全部旧挂单

- 2 → 检查结构与风控条件
- 3 → 若满足条件则生成新网格价格
- 4 → 根据仓位规则分配挂单数量
- 5 → 刷新挂单

生命周期细节：

(1) 订单创建 (Create)

根据网格价格生成挂单：

- 多单挂在 $P_{buy}[k]$
- 空单挂在 $P_{sell}[k]$

挂单附带：

- 数量 (qty)
- 唯一 ID (order.ref)
- 创建时间
- 网格层级

(2) 订单取消 (Cancel)

每根 1m bar 的开头：

取消所有上一个周期未成交的订单。

原因：

- 网格价格每分钟都会计算
- 旧挂单价格不再可信
- 防止旧订单在错误位置成交
- 也能避免重复挂在极端深处

(3) 订单修改 (Amend)

策略不使用 amend，而是：直接取消并重挂

更清晰与干净，也避免不同交易所 amend 不稳定的问题。

5.3 成交处理 (Fill Handling)

当挂单被触发并成交时：

策略进入 **订单回调 (notify_order)** 逻辑：

(1) 记录成交信息

记录：

- 成交时间
- 成交数量
- 成交价格
- 订单层级 (k)
- 当前结构位置
- Fee成本 (按 Maker 费率计算)

写入：

```
grid_trades.csv
```

(2) 更新网格层命中数 (hit_count)

```
hits[level] += 1
```

这是你策略中的关键变量：

- 决定下一轮仓位分配
- 反复被击中的价格 → 仓位变轻
- 未被成交的价格 → 仓位保持较重

(3) 镜像补单 (Symmetric Replenish)

为了维持网格的对称性，需要补仓让网格继续“完整”：

- 多单成交 → 在上方补一个空单
- 空单成交 → 在下方补一个多单

镜像补单的核心逻辑：

代码块

```
1 成交方向: 多 → 镜像方向: 空  
2 成交方向: 空 → 镜像方向: 多
```

补单价格：

- 若当前结构未更新，则补在同一层级距离
- 若区间接近边界，则根据实际可用网格层自动调整

镜像补单让网格在震荡行情中保持价格中性 (Delta-Neutral-ish)。

5.4 RANGE 模式 vs DGT 模式中的挂单行为差异

网格的挂单方式在两个模式中不一样。

5.4.1 RANGE 模式（区间模式）

规则：

- 必须限制在 support ~ resistance 内
- 上下对称挂单
- 使用结构加权 + 衰减进行仓位分配
- 适合区间震荡套利

特点：

双边挂单、仓位对称

5.4.2 DGT 模式（趋势模式）

进入 DGT 必须满足三条件：

2157. 结构突破 (≥ 2 根)

2158. ATR 扩张

2159. 量价确认

从此：

- 网格 不再对称
- 顺势方向网格 更密集且仓位更大
- 逆势方向网格 更稀疏或不挂

例如：

上涨突破 → 多单密集、空单极轻或不挂

- 买入 pullback
- 使用多单加权 $\alpha_{up} > \alpha$
- 卖单仓位衰减 $\tau_{down} \ll 1$

下跌突破 → 空单密集、买单极轻

这种设计使 DGT 模式在趋势方向更稳健。

5.5 挂单数量与频率优化

挂单频率为：

1m 刷新一次

原因：

- 足够快捕捉波动
- 不会过度交易
- 避免 API 频率过高被封
- 足够细粒度控制仓位

挂单数量：

最多 $2 \times \text{max_levels_side} = 20$ 个单

挂单量较低，完全处于各交易所 API 安全范围内。

5.6 防止滑点与冲击成本 (Slippage & Impact Protection)

即使网格使用 Limit-Maker，也必须防止两类风险：

(1) 价格快速穿越网格 (Flash Volatility)

在极端波动时：

- 下单后立刻拉穿价格
- 会被对手价瞬间吃掉

解决方式：

- 每分钟重置挂单
- ATR(50)/ATR(200) 过低 → 冷却
- 越界后进入 BREAKOUT 不再挂单

(2) 挂单价太接近最新价导致 taker 成交

避免：

挂在 0 tick 或太接近盘口

解决方式：

挂单价格必须 \geq 最新价格 $\pm (\text{grid_gap_pct} \times P)$

保证永远不会 Taker 掉。

六、风险控制 (Risk Control Layer)

风险控制层贯穿整个策略，从 RANGE 模式到 DGT 模式，都必须通过该层进行过滤判断。

本策略采用 **五段式风险控制体系**：

249. 突破检测 (Breakout Detection)

250. 假突破过滤 (Fake Breakout Filter)

251. 中轴去库存 (Midline Inventory Reset)

252. 微震荡冷却 (Micro Volatility Cooldown)

253. 方向性仓位上限 (Directional Exposure Cap)

这些子模块共同确保网格策略在波动、高噪声与趋势行情中保持稳健。

6.1 突破检测 (Breakout Detection)

突破检测是整个策略的逻辑分叉点，由它决定：

- 继续 RANGE 模式 (区间)
- 还是进入 BREAKOUT → DGT 模式 (趋势)

本模块必须严格，避免误判。

6.1.1 突破定义 (基于 S/R 结构)

当满足任意一方向：

向上突破：

代码块

```
1 close > resistance + ε
```

向下突破：

代码块

```
1 close < support - ε
```

其中 $ε$ 为动态阈值：

代码块

```
1 ε = ATR(200) × 0.1
```

避免“刺一下”的小跳空，确保突破具有实际意义。

6.1.2 连续 K 线确认 (≥ 2 根)

避免 1 根 K 的假突破。

代码块

```
1 连续2根 close > resistance + ε (或 < support - ε)
```

这一步过滤掉大量错误突破。

6.2 假突破过滤 (Fake Breakout Filter) —— 三重过滤机制

为了避免 DGT 被假突破触发，本策略采用三重过滤：

6.2.1 价格过滤 (Price Confirmation)

满足：

代码块

```
1 close 远离边界 ≥ 0.1% ~ 0.15%
```

数值越大，越不容易被假突破伤害。

6.2.2 波动过滤 (Volatility Confirmation)

核心逻辑：

假突破往往伴随低波动 + 快速刺穿

真突破必须伴随高波动扩张

公式：

代码块

```
1 ATR_short = ATR(50)
2 ATR_long   = ATR(200)
3 ATR_ratio = ATR_short / ATR_long
4 有效突破要求：
5 ATR_ratio ≥ 1.1
```

你目前使用的阈值是 1.0，这里建议升级到 1.1。

6.2.3 成交量过滤 (Volume Confirmation)

假突破通常发生在成交量极低的状态。

公式：

代码块

```
1 volume > SMA(volume, 20) × 1.5
```

这是最有效的确认条件之一。

并非所有突破都需要满足 3 条，可设置为：

突破有效 = 满足条件数 ≥ 2

6.3 中轴去库存 (Midline Inventory Reset)

中轴附近交易价值最低：

- 噪声最大
- 胜率最低
- 价格经常在这里反复打区间

为了避免 inventory 在中轴区域滞留，本策略加入自动去库存：

6.3.1 触发条件

当满足两项：

(1) 价格接近中轴

代码块

```
1 |close - midline| / midline <= mid_band_pct
```

默认：

代码块

```
1 mid_band_pct = 0.1% = 0.001
```

(2) 净仓位偏向过高

代码块

```
1 |net_position| >= exposure_thresh * max_position
```

默认：

代码块

```
1 exposure_thresh = 0.7
```

例如净多达到 70% → 去库存。

6.3.2 动作：部分平仓（Partial Reduce）

算法：

代码块

```
1 reducing_qty = net_position - target_position
```

target_position 可设为 30%~50% 的中性值。

6.3.3 理解逻辑

中轴去库存的两大目的：

- 把利润锁住
- 避免趋势反转时被反杀
- 减少噪声行情中的错误仓位

6.4 微震荡冷却（Micro Volatility Cooldown）

当市场波动极低（极端震荡或休市时），网格无利可图，会无限制被磨损手续费。

为此加入“冷却机制”：

6.4.1 ATR 波动比过滤

计算：

代码块

```
1 ATR(50) / ATR(200) < micro_atr_ratio
```

默认：

代码块

```
1 micro_atr_ratio = 0.6
```

当短期波动不足以支撑网格套利 → 暂停挂新单。

6.4.2 冷却时的行动

- 保留当前挂单（不主动取消）
- 不创建新的网格层级
- 等波动恢复后自动恢复 RANGE 或 DGT 状态

6.5 方向性仓位上限 (Directional Exposure Cap)

网格策略最大风险：

趋势中大量单边挂单被吃 → 仓位累积过重 → 趋势反转爆仓

为避免此风险，加入方向性仓位上限：

6.5.1 最大净仓位限制

代码块

```
1 |net_position| <= exposure_limit * max_position
```

默认：

代码块

```
1 exposure_limit = 0.5 ~ 0.7
```

例如，当总仓位 1.0 BTC 时，最多允许持有 0.5 BTC 多头或空头。

6.5.2 超限 → 自动减仓

如果超限，则自动减仓到目标水平。

6.6 DGT 环境下的额外风险控制

DGT 虽然能让网格在趋势中“不死”，但必须加入：

6.6.1 连续突破保护 (Multi-Break Protection)

当市场剧烈趋势：

连续三次突破 → 不要三次都重置。

应当：

代码块

```
1 设定 reset_cooldown = N 根 K线
```

例如：

代码块

```
1 reset_cooldown = 5 ~ 10 分钟
```

避免连续重置导致网格过度追价。

6.6.2 DGT 趋势反转检测

在 DGT 中必须特别关注：

若反方向出现强突破 → 立即削减仓位 → 返回 RANGE

确保策略不会从趋势的顶或底开始“反向网格”。

6.6.3 滑点保护 (DGT 特有)

突破后立即开网格，可能遇到：

- 巨大瞬时波动
- 交易深度不足
- 大量订单排在盘口后
- 挂单瞬间被 taker 吃掉

解决：

代码块

```
1 挂单间距 ≥ grid_gap_pct × 1.5 during first reset
```

确保 DGT 初次挂单远离噪声区域。

6.7 RANGE / BREAKOUT / DGT 风控状态机

最终风险控制系统以状态机形式运行：

代码块

```
1      +-----+
2      |      RANGE 模式      |
3      +-----+
4      | 假突破 or 冷却
5      v
6      [保持 RANGE 挂单] |
7      | 真突破确认 (>=2)
8      v+-----+
9      | BREAKOUT 模式   |
10     +-----+
11     | ATR>1.1 & 量价确认
12     v+-----+
13     |      DGT 模式      |
14     +-----+
15     |
16     反方向突破 or 冷却 or 波动降级
17     v
18     返回 RANGE
```

这是策略的重大升级点，让你能在：

- 震荡 → 网格吃套利
- 趋势 → DGT 继续赚钱
- 反转 → 风控保护
- 低波动 → 冷却
- 结构恢复 → 回到区间

形成一个完整的闭环。

七、策略执行流程 (Execution Flow)

本策略在运行中分为 **初始化阶段 (Init)** 与 **主循环阶段 (Next)**。

执行逻辑采用 **状态机 (State Machine)** 驱动，核心状态包括：

- **RANGE** (正常区间网格)

- **BREAKOUT** (突破等待确认)
- **DGT** (趋势动态网格)
- **COOLDOWN** (低波动冷却)

所有逻辑都是围绕结构、波动与趋势稳稳地“顺势”，避免未来泄露。

7.1 初始化流程 (Initialization Phase)

策略运行前初始化以下内容：

7.1.1 加载多周期结构数据 (S/R Source)

读取 SR 时间框架（如 4h / 1d / 15m）的历史数据：

- 默认需要过去 90 天 (4h) 或 365 天 (日线)
- 计算 pivot_high/low(20,20)
- 应用 shift(20) 确认机制
- 生成 support / resistance / midline

该数据 与回测时间范围独立。

7.1.2 加载交易时间框架 (1m Data)

读取主执行时间框架 1m 数据：

- 时间范围： `backtest_days`
- 必须包含： `open`、`high`、`low`、`close`、`volume`
- 与 SR 数据通过 `merge_asof` 合并，使每根 1m bar 都知道当时已经确认的 S/R 值

7.1.3 初始化内部变量

初始化策略内部的关键数据结构：

代码块

```
1 state = RANGE
2 hit_count = dict()    # 记录每个网格价格的成交次数
3 breakout_counter = 0
4 reset_cooldown = 0    # DGT 重置冷却计数器
5 active_orders = []    # 当前挂单
6 net_position = 0
7 inventory_log = []
```

7.1.4 初始化风控模块

提前计算：

- ATR(50) / ATR(200)
- volume MA(20)
- micro volatility ratio
- breakthrough epsilon

7.1.5 输出初始化日志

包括：

- SR 来源与时间范围
- 回测日期区间
- 初始状态 (RANGE)
- 初始 S/R 结构

7.2 主循环逻辑 (Main Next Loop)

策略对每根 1m K 线执行完整逻辑：

代码块

```
1 for each bar:run risk_control()  
2     run state_transition()  
3     run grid_generation()  
4     run order_execution()
```

下面分章节详细解释。

7.3 状态机：核心流程 (State Machine)

这是整个策略的心脏。

状态机控制策略何时：

- 开启 / 停止网格
- 重置中心价格
- 切换 DGT 模式
- 暂停因冷却
- 等待突破确认

状态机共有四种状态：

7.3.1 RANGE 状态（区间网格模式）

特点：

- 使用固定 S/R 区间进行网格
- 网格中心为 midline (或 last_price)
- 仓位偏向受控
- 主要吃震荡套利

流程：

代码块

```
1 if price 位于 [support, resistance] 且无冷却:  
2     构建网格  
3     执行边缘加权  
4 else if 发生疑似突破:  
5     进入 BREAKOUT 状态
```

7.3.2 BREAKOUT 状态（突破确认模式）

作用：等待确认是否进入真正趋势。

流程：

代码块

```
1 记录突破方向  
2 breakout_counter += 1  
3 if breakout_counter >= 2 and 波动确认 and 成交量确认:  
4     进入 DGT 状态  
5 elif price 回落区间:  
6     返回 RANGE 状态
```

突破确认机制在第 6 章已写，这里不重复。

7.3.3 DGT 状态（趋势动态网格模式）

这是策略的“高级形态”。

特征：

- 动态计算 新中心价格 (Dynamic Grid Center)

- 方向性加权增强（顺势加码）
- 使用更宽的初始网格距离
- 允许趋势扩张中更深层挂单
- 自动跟踪趋势波动（DGT reset）

主流程：

代码块

```

1 if reset_cooldown > 0:
2     reset_cooldown -= 1
3 else:
4     如果价格进一步远离旧中心：
5         执行 DGT 重置中心 (dynamic center reset)
6         reset_cooldown = cooldown_period

```

趋势反转检测：

代码块

```

1 如果反方向出现 ≥2 根结构突破：
2 减仓
3 返回 RANGE 模式

```

7.3.4 COOLDOWN 状态（低波动冷却模式）

当放量不足 → 不开网格 → 防手术刀行情磨手续费。

触发：

代码块

```
1 ATR(50)/ATR(200) < 0.6
```

行为：

代码块

```

1 暂停挂新网格单
2 不取消已挂的单
3 如果波动恢复 → 返回原状态 (RANGE 或 DGT)

```

7.4 每根 Bar 的完整执行顺序 (Pseudo Code)

下面是策略最核心的部分，定义 next() 的逻辑。

7.4.1 第 1 步：读取当前数据

代码块

```
1 price = close
2 support, resistance, midline
3 atr_short, atr_long, atr_ratio
4 volume, volume_ma
```

7.4.2 第 2 步：风险控制层过滤

代码块

```
1 if 突破检测:
2     state → BREAKOUT 或 DGT
3     清空挂单return
4
5 if 微震荡冷却:
6     state → COOLDOWN
7     keep orders; return
```

7.4.3 第 3 步：状态机逻辑判断

代码块

```
1 if state == RANGE:
2     routine_range()
3
4 if state == BREAKOUT:
5     routine_breakout()
6
7 if state == DGT:
8     routine_dgt()
9
10 if state == COOLDOWN:if 波动恢复:state → RANGE 或 DGTreturn
```

7.4.4 第 4 步：网格生成 (Grid Generation)

生成上下网格价格：

几何网格 (Geometric Grid)

代码块

```
1 P_up[k] = center * (1 + g)^k  
2 P_down[k] = center * (1 - g)^k
```

过滤：

- 仅保留落在 S/R 内的价格
- 若 DGT 模式：允许价格超区间一定比例（动态趋势容忍区）

7.4.5 第 5 步：仓位分配 (Edge Weighting + Decay)

1) 边缘加权

代码块

```
1 weight_long = ((resistance - p)/(resistance - support))^α  
2 weight_short = ((p - support)/(resistance - support))^α
```

2) 命中衰减

代码块

```
1 weight[i] *= exp(-hit_count[i] / decay_k)
```

3) 归一化

7.4.6 第 6 步：挂单刷新 (Order Placement)

每根 bar：

代码块

- 1 取消所有旧订单
- 2 按照权重生成新订单
- 3 记录 order.ref

在 DGT 中：

- 初次挂单扩大间距
- 趋势越强 → 网格可适当加深

7.4.7 第 7 步：订单回调处理 (notify_order)

代码块

```
1 if orderfilled:  
2     hitcount[price] += 1  
3     更新仓位 net_position  
4     DGT 模式下 → 可执行镜像补位  
5     记录日志
```

7.5 执行流程图 (Execution Flow Chart)

代码块

```
1 +-----+  
2 |       初始化 INIT      |  
3 +-----+  
4 |  
5 v  
6 +-----+  
7 |   每根 1m K线开始 next()  |  
8 +-----+  
9 |  
10 +-----+-----+  
11 | Risk Control Filtering |  
12 +-----+-----+  
13 |  
14 +-----+-----+ 状态机切换 -----+  
15 | RANGE | BREAKOUT | DGT | COOLDOWN |  
16 +-----+-----+  
17 |  
18          根据状态执行对应流程  
19 |  
20 +-----+-----+ Grid Generation -----+  
21 | 几何网格 → S/R过滤 → DGT扩展 → 边缘加权 → 衰减 → 分配 |  
22 +-----+-----+  
23 |  
24          挂单刷新 (cancel + place)  
25 |
```

```

26          +-----+
27          |       notify_order      |
28          +-----+
29          |
30          记录成交 / 更新统计
31          |
32          v
33          下一根 K线

```

八、核心参数设计 (Parameter Design)

本策略的参数体系由五大核心分组构成：

- [304. 结构参数 S/R Layer](#)
- [305. 网格参数 Grid Layer](#)
- [306. 仓位与加权参数 Position / Weight Layer](#)
- [307. 风险控制参数 Risk Layer](#)
- [308. DGT 动态趋势参数 DGT Layer](#)

每组参数都对应策略的一层逻辑，通过模块化设计可独立优化。

8.1 结构参数 (S/R Structure Layer)

结构参数用于计算支撑/阻力区间，是整个策略的“地图”。

8.1.1 多时间框架结构来源 (SR Timeframe)

参数：

参数名	默认值	说明
sr_timeframe	"4h"	结构使用的时间框架 (4h / 1d / 15m)
sr_days	90	S/R 来源数据的历史天数
min_sr_bars	500	最少 bars 数量 (防止数据不足)

推荐配置：

- **4h / 90天 (默认) → 最平衡**
- **1d / 365天 → 趋势市场、长期结构**

- 15m / 30天 → 高频震荡结构
-

8.1.2 Pivot 结构参数

参数名	默认值	说明
lookback_perio d	20	pivot 高低点左右窗口大小
pivot_source	close	可选 close / high / low
pivot_confirm_ shift	20	+20 根确认机制，避免未来泄露

核心逻辑：

代码块

```
1 pivot_low = pivot(low, L=20, R=20)
2 confirmed_low = pivot_low.shift(20)
```

所有结构参数通过 `merge_asof` 合并到 1m 数据。

8.1.3 区间有效性参数

参数名	默认值	说明
min_range_pct	0.20%	S/R 最小宽度
range_valid	自动	当 support < resistance 才有效

8.2 网格参数 (Grid Layer)

网格参数定义「网格的形状」，并直接决定你的收益、成功率与稳定性。

8.2.1 网格间距 (Grid Gap)

网格采用几何级数：

```
P_up[k]    = center * (1 + g)^k
P_down[k] = center * (1 - g)^k
```

参数：

参数名	默认值	说明
grid_gap_pct	0.0018 (0.18%)	单层间距
max_levels_side	10	单侧最大层级
grid_expand_dgt	1.5	DGT 模式时扩大首层网格

建议：

- 高频交易 → 0.0008~0.0012
- 日内震荡 → 0.0015~0.003
- 趋势网格 → 0.0025~0.005

8.2.2 网格中心 (Grid Center)

RANGE 模式：

代码块

```
1 center = midline
```

DGT 模式 (动态中心)：

代码块

```
1 center = new_center(price_action, atr)
```

8.3 仓位与加权参数 (Position / Weight Layer)

本策略的仓位分配由两个驱动：

[2166. 结构边缘加权 \(Edge Weighting\)](#)

[2167. 命中衰减 \(Hit Decay\)](#)

8.3.1 边缘加权 (Edge Weighting)

参数名	默认值	说明
alpha	2	权重曲线陡度 (1.5~3.0)

权重函数：

代码块

```
1 w_long = ((resistance - p)/(resistance - support))^\alpha
2 w_short = ((p - support)/(resistance - support))^\alpha
```

含义：

- 越靠近边界 → 权重越大 → 越应该加仓
- 越靠近中轴 → 权重越小 → 不要交易噪声

8.3.2 命中衰减 (Hit Decay)

参数名	默认值	说明
decay_k	2	成交重复位置衰减系数

公式：

代码块

```
1 weight *= exp(-hit_count / decay_k)
```

功能：

- 防止重复在同一价格来回交易
- 避免过度仓位堆积
- 提升网格“探索性”与“分散性”

8.3.3 总仓位预算

参数名	默认值	说明
order_size	0.05	单侧总仓位预算 (以币计)
max_position	1	整体最大仓位限制

8.4 风险控制参数 (Risk Layer)

风险参数是策略的「生命线」，决定策略在趋势、震荡衰竭、假突破中的存活率。

此模块已在第六章详述，这里给出参数定义。

8.4.1 突破过滤参数

参数名	默认值	说明
breakout_epsilon_mult	0.1	支撑/阻力阈值比例 ($ATR \times 0.1$)
breakout_atr_ratio	1.1	ATR 确认阈值
breakout_volume_mult	1.5	量能确认阈值

突破确认必须满足三者中的 2 项以上。

8.4.2 中轴去库存参数

参数名	默认值	说明
mid_band_pct	0.10%	进入中轴区域
exposure_threshold	0.7	仓位偏向阈值
rebalance_target	0.4	去库存后的目标仓位比例

8.4.3 微震荡冷却参数

参数名	默认值	说明
micro_atr_ratio	0.6	ATR 低于该比率暂停下单

8.4.4 方向性仓位上限 (Exposure Cap)

参数名	默认值	说明
exposure_limit	0.5	最大偏向仓位 (多或空)

超过自动减仓。

8.5 DGT 动态趋势参数 (DGT Trend Layer)

DGT 是策略的核心创新点，参数设计必须严谨。

8.5.1 趋势确认条件

参数名	默认值	说明
dgt_breakout_bars	2	连续突破根数
dgt_atr_ratio	1.1	波动确认阈值
dgt_volume_mult	1.3	量能确认阈值

8.5.2 动态中心重置 (DGT Reset)

参数名	默认值	说明
dgt_reset_threshold	$ATR(200) \times 0.5$	超过才重置中心
reset_cooldown_n_period	5 bars	避免连续重置
dgt_initial_grid_mult	1.5	重置后的第一轮 网格加大间距

核心逻辑：

代码块

```

1 if |price - center| > 阈值:
2     center = price
3     扩张初始网格
4     reset_cooldown 开始计时

```

8.5.3 趋势反转检测

参数名	默认值	说明
dgt_reverse_bars	2	反方向突破确认
dgt_reduce_factor	0.5	反转时削减仓位比例

8.6 参数分组总结（机器可读版）

可直接用于 config.yaml:

代码块

```

1   sr:
2     timeframe: "4h"
3     days: 90
4     min_bars: 500
5     pivot_lookback: 20
6     pivot_shift: 20
7
8   grid:
9     gap_pct: 0.0018
10    max_levels: 10
11    expand_mult_dgt: 1.5
12
13  position:
14    order_size: 0.05
15    max_position: 1.0
16    edge_alpha: 2.0
17    decay_k: 2.0
18
19  risk:
20    breakout_epsilon_mult: 0.1
21    breakout_atr_ratio: 1.1
22    breakout_volume_mult: 1.5
23    mid_band_pct: 0.001
24    exposure_thresh: 0.7
25    rebalance_target: 0.4
26    micro_atr_ratio: 0.6
27    exposure_limit: 0.5
28
29  dgt:
30    breakout_bars: 2
31    atr_ratio: 1.1
32    volume_mult: 1.3
33    reset_threshold_atr_mult: 0.5

```

```
34     reset_cooldown: 5  
35     initial_grid_mult: 1.5  
36     reverse_bars: 2  
37     reduce_factor: 0.5
```

九、仓位管理设计

(Position Management Layer)**

仓位管理是本策略的第二核心（第一核心是结构 + DGT）。

它负责：

- 控制风险
- 平滑收益
- 降低极端行情中的仓位暴露
- 管理网格累计仓位
- 在趋势中顺势扩张 (DGT)
- 在盘整中保持均衡 (RANGE)

整个仓位管理体系由 **四大模块** 构成：

430. 仓位框架 (Position Framework)

431. 区间模式仓位逻辑 (RANGE Positioning)

432. 趋势模式仓位逻辑 (DGT Positioning)

433. 中枢库存管理 (Inventory Management)

9.1 仓位框架 (Position Framework)

本策略采用“网格可控仓位体系”，包含三个层次：

9.1.1 单层仓位 (Layer Size)

每一层的挂单数量来自：

```
layer_qty = total_side_budget * weight_i
```

- `total_side_budget` = `order_size`
- `weight_i` = 加权后的第 i 层权重

特点：

- 越靠近边界 → 越大：低风险、高胜率
- 越靠近中轴 → 越小：高噪声区减少交易

- 越被触发次数多 → 越小：通过衰减排除“假区间”

9.1.2 单侧最大仓位 (Side Exposure)

默认：

代码块

```
1 order_size = 0.05
```

表示——

无论多少网格层，单侧最多 0.05 单位（币）的仓位。

避免趋势中网格层数多 → 仓位无限增加。

9.1.3 总体最大仓位 (Total Exposure Cap)

默认：

代码块

```
1 max_position = 1.0
```

这是绝对仓位上限。

无论多空如何叠加，永远不能超过此值。

9.1.4 方向性限制 (Directional Limit)

代码块

```
1 |net_position| <= exposure_limit * max_position
```

默认：

代码块

```
1 exposure_limit = 0.5
```

意义：

- 防止网格在趋势中被“吃穿”导致单边重仓
- 趋势不明时保持中性敞口

9.2 区间模式仓位逻辑 (RANGE Positioning)

区间行情中仓位逻辑的目标是：

- 低吸高抛
- 不预测方向
- 保持中性（或轻微偏向）
- 避免在中轴处重仓

因此 RANGE 模式的仓位分布是：

代码块

- 1 靠近支撑：多单多
- 2 靠近阻力：空单多
- 3 靠近中轴：两边都少

9.2.1 边缘加权带来的自适应仓位

加权公式（第 8 章已定义）：

- 靠近支撑 → 多单权重爆炸式增加
- 靠近阻力 → 空单权重增加
- 中轴 → 双边几乎无仓

这种分布天然符合区间交易的数学优势：

- R 端开空 → 高胜率
- S 端开多 → 高胜率
- M 端 → 噪声大 → 自动降低交易频率

这是你的策略 **不容易爆炸的核心原因之一**。

9.2.2 命中衰减控制重复交易

如果某一层价格连续被来回触发

- 衰减使该位置越来越小
- 自动减少重复、无效的噪音交易

```
weight *= exp(-hit_count / decay_k)
```

意义：

- 避免过度频繁交易
- 自适应避免“失效区间”

9.2.3 区间仓位中性原则 (Neutral Inventory Principle)

区间行情不需要方向性赌注，必须遵守：

```
net_position ≈ 0
```

策略的中轴去库存（第 6 章）正是为了维持这一状态。

9.3 趋势模式仓位逻辑 (DGT Positioning)

趋势模式是 DGT 最大的亮点之一。

当检测到真突破，你不再进行区间对冲，而是：

- 顺势偏向
- 动态重置网格中心
- 让网格叠加趋势收益
- 同时用轻量仓位控制风险

9.3.1 趋势偏向逻辑 (Trend Bias Positioning)

趋势确认后，允许偏向多或空：

- 上涨趋势：多 > 空
- 下跌趋势：空 > 多

偏向程度受：

代码块

```
1 exposure_limit  
2 order_size  
3 alpha
```

共同控制，确保：

- 顺势赚更多
- 但不会重仓到危险程度

9.3.2 DGT 重置后仓位扩展方式

重置中心后：

[2543. 网格间距扩大 \(初次 DGT\)](#)

[2544. 远离原区间的部分允许轻仓延伸](#)

[2545. 越接近当前趋势加速位置 → 越轻仓](#)

2546. 越接近趋势回调区 → 可以加大仓位

形成一个“趋势友好型的网格结构”：

趋势方向 → 小仓

回调位置 → 大仓（风险更小）

这是与普通网格完全不同的优势。

9.3.3 DGT 的仓位保护 (Trend-Safe Inventory)

如果趋势反转：

- 立即削减仓位
- 触发回到 RANGE 模式
- 禁止继续扩张网格（进入 reset 冷却）

这是 DGT 不容易翻车的核心保护逻辑。

9.4 中枢库存管理 (Inventory Management)

你在第 6 章提出“中轴去库存”，现在我们系统化写成完整的“库存管理层”。

目标：

- 不让 net_position 偏离太大
- 避免“方向性爆仓风险”
- 避免中轴噪声造成连续误判
- 保证区间内仓位健康

库存管理由三部分组成：

9.4.1 中轴去库存 (Midline Reset)

当价格靠近 midline：

代码块

```
1 |net_position| >= exposure_thresh * max_position
```

→ 启动去库存：

代码块

```
1 reduce_to_rebalance_target
```

例如：

- 当前 net_position = 70%
- 目标 = 40%
- 减仓 = 30%

极大减少“噪声中心持仓”。

9.4.2 趋势反转去库存 (Trend Reversal Reduce)

DGT 中，如果反方向出现突破：

```
reduce_position(dgt_reduce_factor)
```

例如：

- 持有多仓 0.6
- dgt_reduce_factor = 0.5
- → 平掉 50% 多单

9.4.3 极端风险削减 (Emergency Deleveraging)

若仓位暴露过大（例如 >50%）且：

```
价格快速超出网格 n 倍 ATR
```

则执行紧急减仓：

```
force_reduce_to_neutral
```

确保自动从高风险区域退出。

9.5 仓位管理模块总结

仓位管理层的设计目标是：

让网格策略在震荡中吃反复套利，在趋势中顺势赚钱，并在噪音与反转中避免死亡。

通过以下五个核心机制实现：

- 1) 边缘加权：仓位向 S/R 边界集中 → 更高胜率
- 2) 命中衰减：减少重复无效交易 → 更低成本
- 3) 区间仓位中性：不赌方向 → 更稳
- 4) 趋势仓位偏向：顺势可扩张 → 能吃趋势大钱
- 5) 库存去化 & 方向上限：避免方向性爆仓 → 更安全

整个仓位结构符合：

- 震荡 → 多次小盈利
- 趋势 → DGT 叠加赢（真正赚钱）
- 假突破 → 风控保护
- 中轴 → 自动降低仓位
- 极端波动 → 自动减仓

十、订单生成与执行逻辑（Order Execution Rules）

本策略使用「限价单 + 网格挂单刷新」方式执行交易。

执行结构包含四个部分：

250. 网格订单生成（Grid Generation）
251. 订单挂单刷新机制（Order Refresh Cycle）
252. 订单成交逻辑（Fill Logic）
253. 订单撤销与保护（Cancel & Safety Filters）

并且对 RANGE 与 DGT 两种模式使用不同执行参数。

10.1 网格订单生成逻辑（Grid Generation）

网格订单基于两要素：

538. 网格中心（center）
539. 网格间距（gap_pct）

10.1.1 网格定价公式（几何级数）

上行网格（卖单）：

代码块

```
1 P_sell[k] = center * (1 + gap_pct)^k
```

下行网格（买单）：

代码块

```
1 P_buy[k] = center * (1 - gap_pct)^k
```

特点：

- 间距随层级增大呈“等比例扩散”
- 越远离中心 → 越容易被成交
- 越靠近中轴 → 更密集（但权重更小）

10.1.2 区间过滤（仅 RANGE 模式）

过滤到有效区间：

代码块

```
1 support <= P_buy[k] <= resistance  
2 support <= P_sell[k] <= resistance
```

如果一侧全部不在区间，则不挂该侧。

10.1.3 DGT 扩展网格（趋势模式）

DGT 能突破区间结构，因此允许“超区间”部分挂单：

允许最多 `extend_ratio × ATR(200)` 的延伸范围

默认：

`extend_ratio = 0.5`

目的：

在趋势中扩大可捕捉到的「回调套利带」。

10.1.4 网格层级裁剪

单侧最大层级：

代码块

```
1 max_levels_side = 10
```

如果区间狭窄，只保留落入范围内的价格。

10.2 网格挂单刷新机制（Order Refresh Cycle）

网格策略的关键是：

每根 1m bar 全量重建订单（全撤全挂）。

流程顺序：

1346. 取消当前所有挂单

1347. 通过最新价格与结构重新生成网格

1348. 计算权重与分配仓位

1349. 下新的限价订单

所有挂单必须与当前结构匹配，不得沿用旧订单。

10.2.1 撤单顺序 (Cancel First)

代码块

```
1 for o in active_orders:  
2     if o.status == Pending:  
3         o.cancel()  
4 clear active_orders
```

这是所有网格策略的标准操作，避免“旧网格残留”。

10.2.2 下单顺序 (Recreate New Orders)

下单必须遵循顺序：

1660. 先创建 BUY 订单 (下方)

1661. 再创建 SELL 订单 (上方)

原因：

- BUY 订单靠前提交 → 优先获得更高深度
- SELL 订单放后面减少提前触发风险

10.2.3 下单类型

全部是限价单：

代码块

```
1 order_limit(price, qty)
```

默认 maker-only。

火币/币安/OKX 都有：

代码块

```
1 post_only = true
```

确保不被当场吃掉造成 taker 手续费。

10.2.4 DGT 模式下的挂单调整

第一次进入 DGT 模式或 DGT reset 时：

- 首轮网格扩大 spacing (更宽网格)
- 增大第一层的 gap_pct
- 减少过度密集挂单导致被连续扫穿

例如：

代码块

```
1 gap_pct_dgt = gap_pct * initial_grid_mult  
2 initial_grid_mult = 1.5 ~ 2
```

这样能够避免趋势中因为距离过近导致连续吃单 → 手续费损失加剧。

10.3 订单成交逻辑 (Fill Logic)

订单成交事件通过回调函数 (notify_order / on_fill) 处理。

10.3.1 成交触发的动作

当订单成交：

2330. 记录成交价格、数量
2331. 更新 `net_position`
2332. 更新 `hit_count[price]`
2333. 如果是 RANGE 模式 → 镜像补仓机制 (mirror refill)
2334. 如果是 DGT 模式 → 只做 “趋势友好型补单”
2335. 写入交易日志、成交记录

10.3.2 镜像补单 (Mirror Refill)

这是 RANGE 模式的核心：

若某位置被成交 (比如 BUY) : 需要在对应的对称位置放一个 SELL

示意：

```
1 BUY at support+1% → SELL at resistance-1%
```

这样确保区间套利的完整性：
每一次成交 → 都伴随一次潜在的套利闭合机会。

10.3.3 DGT 模式下的成交行为

DGT 中不采用“对称补单”，因为趋势中对称位置可能永远回不来。

改为：

- 趋势方向：轻量补单
- 回调方向：允许较大仓位（靠边界位置）
- 保持整体仓位“不超限，也不丢失趋势机会”

10.3.4 Deal-By-Deal 记录与 hit_count

成交后：

代码块

```
1 hit_count[price] += 1
```

作用：

- 通过衰减机制减少重复交易
- 动态降低“失效网格线”的权重
- 自适应区间行为

10.4 订单撤销与执行保护（Cancel & Safety Filters）

策略必须具备「坏行情下安全退出订单」的能力。

以下四类情况必须撤单：

10.4.1 突破检测（Breakout Detected）

当满足突破条件：

代码块

```
1 cancel_all_orders()  
2 state = BREAKOUT  
3 stop_trading
```

避免网格继续在“真趋势”腿上被来回穿刺。

10.4.2 区间无效 (Range Invalid)

如果：

- support 或 resistance 为 NaN
- support \geq resistance
- price 不在区间内

必须立即：

代码块

```
1 cancel_all_orders()  
2 no new orders
```

10.4.3 微震荡冷却 (Micro Cooldown)

当 ATR_ratio < 0.6:

- 不撤单
- 但不挂新单
- 保持已有挂单待成交（或不成交也无妨）

目的是避免：

- 手术刀行情
- 没有波动却高频挂单 → 手续费磨损

10.4.4 DGT 重置冷却 (Reset Cooldown)

reset 后一定周期内（如 5 根）不能再次 reset:

```
if reset_cooldown > 0: do not rebuild center
```

避免连续在趋势高波动期被来回重置。

10.5 订单执行流程图

代码块

```
1 +-----+  
2 | 1m Bar arrives (next) |  
3 +-----+
```

```

4
5
6 +-----+
7 | Risk Filters (Breakout / ATR / SR) |
8 +-----+
9
10    condition failed? -----> cancel all → return
11
12    v+-----+
13    | State Machine |
14 +-----+
15
16        RANGE → 生成标准网格
17        BREAKOUT → 等待确认
18        DGT → 生成趋势网格 |
19    v+-----+
20    | Cancel all orders |
21 +-----+
22
23    v+-----+
24    | Generate Grid Levels |
25 +-----+
26
27    v+-----+
28    | Compute Weights & Allocate |
29 +-----+
30
31    v+-----+
32    | Place limit orders |
33 +-----+
34
35    v+-----+
36    | notify_order callback |
37 +-----+
38
39        vUpdate hit_count, Position|
40        vEND → Next Bar

```

c回测框架与输出 (Backtesting & Metrics)

本策略采用 **事件驱动式 1m 回测框架**, 支持:

- 多周期数据合并 (4h/1d → 1m)
- 动态中心 (DGT)
- 结构确认 (pivot shift)

- 网格订单刷新
- 成交回调
- 费用与滑点建模

回测输出分为三层：

358. 交易级输出 (Trade Level Metrics)

359. 网格级输出 (Grid Level Metrics)

360. 策略级输出 (Strategy Level Metrics)

11.1 回测框架设计 (Backtest Engine Architecture)

采用事件驱动式 Backtest Flow：

代码块

```
1 Bar arrives → Risk Filter → State Machine  
2 → Grid Regeneration → Order Execution → Update Position
```

11.1.1 数据处理 (Data Handling)

包括两种数据来源：

(1) 结构时间框架数据 (S/R TF)

- 4h、1d 或 15m
- 历史长度独立于回测范围
- 用于 pivot(20,20) + shift(20)

(2) 1m 执行数据

- 用于驱动 next()
- 按真实市场顺序回放

合并方式：

代码块

```
1 merge_asof 方向 backward
```

确保 1m Bar 只看到已经确认的结构。

11.1.2 手续费与滑点模型 (Fees & Slippage)

本策略是：

- 限价单 = Maker-only
- 无滑点 (默认)

可设置手续费：

代码块

```
1 maker_fee = 0.0002    (0.02%)
```

可选模拟滑点：

代码块

```
1 slippage_pct = 0.00005  (0.005%)
```

但网格策略一般使用 maker-only，不加滑点也可。

11.1.3 订单回调设计 (Fill Callback)

每笔订单触发 `notify_order()`：

- 更新 `net_position`
- 更新 `hit_count`
- 更新 PnL
- 写入 trade-log

全部详细结构化记录到 CSV。

11.2 交易级输出 (Trade-Level Metrics)

每一笔成交都写入交易记录表：

`grid_trades.csv`

字段建议 (专业量化格式)

字段	说明
timestamp	成交时间
side	buy / sell
price	成交价
qty	数量
fee	手续费
grid_level	网格层级 k
weight	当前权重
sr_support	当时的支撑
sr_resistance	当时的阻力
sr_midline	中轴
state	RANGE / BREAKOUT / DGT
hit_count	该价格历史成交次数
position_before	成交前仓位
position_after	成交后仓位

这张表最有价值，因为：

- ✓ 可计算交易路径
- ✓ 可定位策略问题
- ✓ 可看出网格效率
- ✓ 可看出突破行为
- ✓ 可验证 DGT 逻辑

11.3 网格级输出 (Grid-Level Metrics)

用于衡量每一层网格的贡献度。

输出 1：网格层级使用频率 (Grid Usage Frequency)

记录每一层的：

- 成交次数
- 成交金额
- PnL 贡献
- 成交位置（上下边界靠近度）

示例表格：

层级 k	价格	成交次数	多单次数	空单次数	PnL	hit_count
------	----	------	------	------	-----	-----------

用途：

- 判断 gap_pct 是否合理
- 看是否有过度密集交易
- 分析边缘加权效果

输出 2：命中衰减热力图（Hit Decay Heatmap）

展示：

- 某一层是否被不断触发
- 是否存在“假区间”导致重复交易
- 衰减机制是否有效降低重复交易的权重

输出 3：网格对称性分析（Buy/Sell Balancing）

用于验证：

- 镜像补单是否有效
- 是否存在单边累积风险
- 仓位是否偏向某个方向过多

11.4 策略级输出（Strategy-Level Metrics）

策略级输出用于整体评价策略表现。

主要包括：

- [2669. 收益相关](#)
- [2670. 风险相关](#)
- [2671. 效率相关](#)
- [2672. 模式分布相关（RANGE vs DGT）](#)

11.4.1 收益类指标（Return Metrics）

指标	说明
总收益率	最终余额 / 初始余额
年化收益 (APR)	按365天年化
夏普比率	风险调整收益
盈亏比	Gross Profit / Gross Loss
最大单笔盈利/亏损	极端点测试
Average PnL per trade	平均单笔收益
Win Rate	胜率，多用于判断网格效率

11.4.2 风险类指标 (Risk Metrics)

指标	说明
最大回撤 (Max Drawdown)	策略最大资金回撤
最大仓位暴露 (Max Exposure)	单侧或总仓位最大值
左尾风险 (Left Tail Risk)	DGT 反转时导致的损失
趋势杀伤 (Anti-Trend Loss)	区间向趋势切换时的损失

11.4.3 效率类指标 (Efficiency Metrics)

重点衡量网格效率与 DGT 成效：

指标	说明
每日网格成交次数	回测周期中的平均频率
单笔手续费成本	平均手续费
费率占总收益比	#NAME?
网格效率 (Grid Efficiency)	#NAME?
DGT收益贡献比 (DGT Contribution)	趋势模式贡献 / 总收益

11.4.4 模式分布类指标 (Mode Distribution)

策略运行时间分布：

状态	占比
RANGE 时间	%
BREAKOUT 时间	%
DGT 时间	%
COOLDOWN 时间	%

这可以告诉你：

- 市场真正震荡时间有多少？
- 趋势模式占比是否影响收益？
- COOLDOWN 是否过度触发？

11.5 可视化输出 (Visual Outputs)

可视化用于更直观展示策略行为。

建议输出以下图表：

图 1：净值曲线 (Equity Curve)

- 主图放 Equity
- 副图放总仓位与方向性敞口

图 2：价格 + 网格线回放图 (Grid Overlay Chart)

主图绘制：

- price
- support / resistance / midline
- 网格线（不同模式用不同颜色）
- RANGE 区间背景
- DGT 重置位置标注

可看到：

- DGT 是否正确重置
- 网格是否过密

- 结构是否过早/过迟确认

图 3：DGT 状态分布图

展示趋势段落，以色块标注：

- 黄色：BREAKOUT
- 红色：DGT
- 灰色：COOLDOWN

一眼看出策略状态机是否合理。

图 4：网格效率图（Grid Efficiency）

展示每层网格的收益贡献：

k=1 → 贡献多少

k=2 → 贡献多少

...

可发现哪些层级最赚钱，哪些层级是噪音。

图 5：hit_count 热力图

查看哪些价格区域被来回打：

- 如果很集中 → 区间太窄
- 如果很散 → 区间预测良好
- 如果只有一两个点特别热 → 是否结构失效？

11.6 回测输出文件结构

建议输出以下文件：

代码块

```

1  results/
2  |
3  └── summary.json      # 指标汇总
4  └── equity_curve.csv  # 净值序列
5  └── grid_trades.csv   # 完整交易记录
6  └── position_log.csv  # 每根bar的仓位
7  └── mode_distribution.csv# RANGE/DGT时间分布
8  |
9  └── plots/
10    └── equity_curve.png
11    └── grid_overlay.png

```

```
12      └── grid_efficiency.png  
13      └── dgt_segments.png  
14      └── hit_heatmap.png
```

这些都是开发者、你、量化同行都会非常喜欢的结构。

11.7 回测性能评估流程 (Validation Pipeline)

回测的正确顺序：

5009. 先看 net value
5010. 再看 drawdown
5011. 再看 grid trades (效率)
5012. 再看 DGT 效果 (趋势)
5013. 再看仓位与风险敞口
5014. 最后再看参数敏感性

确保你的策略没有：

- 中轴磨损过大
- 趋势扩张时吃满亏损
- 假突破造成重置频繁
- DGT 重置过度
- 仓位失控或过度偏向

十二、敏感性分析与优化 (Sensitivity & Optimization)

本策略具有多个模块 (S/R、Grid、Risk、DGT)，每个模块都包含可优化参数。

参数太多会导致：

- 过拟合
- 曲线拟合
- 历史依赖
- 对未来行情失效

因此本章分为：

364. 关键参数敏感性分析 (Key Sensitivity)
365. 网格参数优化 (Grid Tuning)
366. 结构参数优化 (S/R Tuning)

367. DGT 参数优化 (Trend Tuning)

368. 风险参数优化 (Risk Tuning)

369. 多市场 / 多环境测试 (Market Regime Testing)

370. 稳健性验证 (Robustness)

最终目标：

找到既稳定又能不断适应行情的策略参数结构。

12.1 关键参数敏感性分析 (Key Sensitivity Analysis)

我们先确定哪些参数最影响收益 / 回撤：

高敏感参数 (极需小心)

(稍微调动就可能导致策略灾难)

模块	参数	说明
Grid	grid_gap_pct	最敏感参数，过小→手续费爆炸，过大→少交易
Risk	breakout_epsilon_mult	假突破过滤关键，影响 DGT 触发频率
DGT	reset_threshold_atr_mult	DGT 重置频率关键，如果太低→疯狂重置
Risk	exposure_limit	控制方向性风险，过高极危险
Grid	max_levels_side	超过一定层数会变成「无限摊平」

中敏感参数 (需优化)

模块	参数	说明
Grid	max_levels_side	5~10 都可，需要回测测试
Position	alpha	控制边缘加权力度
Risk	micro_atr_ratio	控制低波动时是否暂停

低敏感参数（可自由调整）

模块	参数	说明
S/R	sr_days	数据范围适度调节影响不大
S/R	lookback_period	15~25 都较稳
DGT	initial_grid_mu lt	稍微变化影响很小

当你未来做自动化优化 (optimizer) 时，应优先优化 **高敏感** 与 **中敏感** 参数。

12.2 网格参数优化 (Grid Tuning)

网格交易中最重要的参数是：

- grid_gap_pct
- max_levels_side

优化方法是：

12.2.1 网格间距扫描 (Gap Scan)

扫描范围：

- 0.0008 → 高频
- 0.0012 → 标准
- 0.0018 → 默认
- 0.0025 → 大震荡
- 0.0035 → 稀疏

针对每个 gap 进行：

指标	含义
Net PnL	总收益
Fee %	手续费占净利比例
Efficiency	网格套利效率
DGT Contribution	趋势段收益贡献

优化目标：

- Fee% 不超过 30%

- Efficiency $\geq 40\%$
- Net PnL 最大化
- 曲线平稳无大跳空

12.2.2 最大层数扫描 (Levels Scan)

扫描:

`5, 7, 10, 12`

过高层数 \rightarrow 无限摊平风险

过低层数 \rightarrow 网格不够密集

建议范围:

`7 ≤ max_levels_side ≤ 10`

12.3 结构参数优化 (S/R Tuning)

S/R 是网格的核心结构来源。

可优化:

- 时间框架 `sr_timeframe`
- 数据范围 `sr_days`
- pivot window `lookback_period`

12.3.1 时间框架测试 (TF Scan)

测试:

- 4h
- 1d
- 15m

结论 (基于经验 + 你策略设计) :

时间框架	适用场景
4h (推荐)	大部分币种、震荡与起势皆可
1d	趋势币，结构更稳定
15m	高频套利，回撤更小但贡献更低

通常：

4h ≈ 最平衡的选项

12.3.2 pivot window 优化

扫描：

15, 20, 25, 30

pivot 越大 = 越稳定

pivot 越小 = 越敏感（假突破增加）

通常：

20 = 黄金值

12.3.3 S/R 数据范围优化 (sr_days)

扫描：

30d, 60d, 90d, 180d

- 太短：结构不稳定
- 太长：“过时结构”会误导策略

建议：

4h → 90天最佳

1d → 365天最佳

12.4 DGT 参数优化 (Trend-Tuning)

DGT 的可靠性决定了：

- 能否吃到趋势收益
- 是否会被假突破反杀
- 是否过度重置

优化 DGT 参数是策略成功的关键之一。

12.4.1 重置阈值扫描 (Reset Threshold)

扫描：

ATR(200) × 0.3

ATR(200) × 0.5 (默认)

ATR(200) × 1.0

效果：

- 0.3 → 频繁重置，手续费增加
- 0.5 → 最平衡
- 1.0 → 重置太少，容易掉队趋势

12.4.2 趋势确认扫描

扫描：

```
dgt_atr_ratio: 1.0, 1.1, 1.2
volume_mult: 1.2, 1.3, 1.5
```

建议：

- 趋势过滤不能太松，否则过度进入 DGT
- 越严格 → 趋势段更干净，但是错过时机

现在的默认：

```
atr_ratio = 1.1
volume_mult = 1.3
```

已经很稳健。

12.4.3 DGT 重置冷却 (reset cooldown)

扫描：

```
3, 5, 8, 12 bars
```

意义：

- 太短 → 重置过密
- 太长 → DGT 不敏捷

5 是黄金区。

12.5 风险参数优化 (Risk Tuning)

风险参数必须极度谨慎，因为：

- 太严 → 不做单
- 太松 → 网格会死

测试哪些风险参数：

12.5.1 假突破过滤优化 (Fake Breakout)

参数：

- breakout_epsilon_mult = ATR × (0.08~0.2)

- `breakout_atr_ratio = 1.0~1.2`
- `breakout_volume_mult = 1.2~2`

一般组合：

- `epsilon_mult` 越大 → 趋势确认越晚
- `atr_ratio` 越大 → 假突破越少
- `volume` 越大 → 趋势越干净

组合建议：

紧：0.08, 1.0, 1.2

稳：0.10, 1.1, 1.3 (默认)

松：0.15, 1.2, 1.5

12.5.2 中轴去库存参数优化

扫描：

`mid_band_pct: 0.08% ~ 0.15%`

`exposure_thresh: 0.6 ~ 0.8`

`rebalance_target: 0.3 ~ 0.5`

建议：

0.1%, 0.7, 0.4

效果最平衡。

12.6 多市场测试 (Multi-Market Testing)

必须用至少 6~10 个币做回测：

大市值：

- BTC
- ETH
- BNB

强波动中市值：

- SOL
- AVAX
- LINK

极高 beta (测试 DGT) :

- NEAR
- APT
- DOGE
- SHIB

测试指标：

- DGT 效率
- RANGE 效率
- 最大回撤
- 网格密度
- 假突破数量
- 每次 DGT 重置的盈利贡献

你的策略属于「结构驱动的网格 + 趋势扩展」，非常适合多市场测试。

12.7 市场环境测试 (Regime Testing)

必须在三种环境测试：

市场环境	目标
震荡	网格套利效率
趋势	DGT 收益贡献
高噪声	假突破过滤能力

具体：

震荡测试：

- ETH/BTC
- BTC/USDT 在 2023 Q2

趋势测试：

- BTC 2023 Q1
- SOL 2023 Q4
- BTC 2024 Q4

高噪声测试：

- DOGE

- SHIB
- APT 2023 Q2

验证策略在所有市场环境下都具备稳健性。

12.8 稳健性验证 (Robustness Testing)

四类验证：

12.8.1 参数扰动测试 (Parameter Perturbation)

把所有参数 $\pm 10\sim 20\%$ 扰动：

- 净值曲线应保持合理稳定
- 不应该出现“突然崩溃”的情形

这是检验策略是否过拟合最强的方法。

12.8.2 前推验证 (Forward Testing)

如果优化区间使用了 30 天数据

必须用后 30 天做 forward 测试。

12.8.3 蒙特卡洛重排测试 (Monte-Carlo Resampling)

对成交顺序随机重排 → 测试极端成交顺序下是否还稳健。

12.8.4 滑点增量测试 (Slippage Stress Test)

在默认滑点基础上增加：

0.01% → 0.02% → 0.05%

每加一次滑点 netPnl 的下降幅度应可控，不应断崖式下跌。

十三、策略优劣势与未来扩展

(Strengths, Weaknesses, Extensions)**

本策略是一个 **结构驱动的动态网格系统 (Structure-Adaptive Dynamic Grid Strategy, v2)**，融合：

- 结构 (S/R)
- 网格 (Grid)
- 趋势动态中心 (DGT)
- 多层风控 (ATR、Volume、Breakout Filtering)

- 仓位管理 (Inventory Control)
- 状态机 (RANGE / BREAKOUT / DGT / COOLDOWN)

这是一个高度综合型系统。

这一章我们总结策略的优势、限制，并为未来扩展提供方向。

13.1 策略优势 (Strengths)

本策略有六大核心优势，使其区别于传统网格策略：

优势 1：多周期结构驱动 (S/R + shift(20))

传统网格 → 完全依赖固定区间

我们的策略 → 依赖 **更高时间框架结构**

结构来自：

- 4h pivot
- 1d pivot
- 15m pivot

通过 shift(20) 实现 **无未来泄露** 和 **机构级结构确认**。

优势：

- 极大降低假区间导致的连续亏损
- 自动适应市场结构
- 区间边界天然具备成交集聚 → 盈利概率更高

优势 2：动态中心网格 (DGT) 避免趋势爆仓

传统网格会在趋势行情中被“穿透爆炸”，因为中心固定。

我们的策略采用：

- **DGT (Dynamic Grid Tracking)**
- 自动重置中心
- 自适应趋势扩张
- 自动控制方向性仓位
- 趋势反转自动减仓

使网格能够：

- **在趋势中赚钱**
- 而不是单纯做震荡

这是整个策略的革命性优势。

优势 3：多层风控机制（机构级）

包含四层保护：

- 1173. 突破检测 (ATR+价格+volume)
- 1174. 假突破过滤 (防止误入趋势模式)
- 1175. 中轴去库存 (避免噪声区域累计仓位)
- 1176. 方向性仓位上限 (Exposure Cap)

这些机制保证策略：

- 不会在极端行情中彻底爆炸
- 不会因为错误趋势判断导致损失扩大
- 不会无脑向一个方向摊平

优势 4：边缘加权 + 命中衰减形成智能网格

你的网格不是“均匀挂单”，而是：

- 越靠近区间边界 → 仓位越大 → 风险更低
- 越靠近中轴 → 仓位越小 → 避免噪音交易
- 越被重复打 → 权重越小 (hit decay)

这种智能分布让网格表现更稳健、更聪明。

优势 5：可视化与可回测性强 (Backtest-friendly)

优秀的策略必须能被：

- 验证
- 回测
- 优化
- 调参

本策略使用：

- merge_asof
- pivot shift
- 状态机
- 结构驱动网格
- 动态中心

所有系统都是 事件驱动 + 未来泄露，非常适合自动化回测。

优势 6：跨市场、跨周期表现稳健

因为结构驱动本质：

- 高频币种 (SOL、APT、OP) → 波动型网格收益高
- 大市值稳定币 (BTC、ETH) → 区间收益稳、DGT 可以吃趋势
- meme 高噪声币 (DOGE、SHIB) → 冷却+过滤避免随机噪声交易

策略是“市场环境适应型”，不会因为市场变化而突然失效。

13.2 策略弱点 (Weaknesses)

所有策略都有弱点，承认并加强它们更重要。

弱点 1：结构识别依赖 pivot 机制

pivot 有天然缺陷：

- 无法捕捉“结构快速变化”
- 需要至少 20 根确认延迟
- 高频环境下响应速度可能不够快

可能导致：

- 区间结构未更新 → 网格变弱
- 趋势来临稍有延迟 → DGT 稍后进入

解决（未来扩展里会提到）：

- 引入密集度结构识别 (volume clustering)
- 引入订单流结构
- 引入 AI 结构分类模型

弱点 2：假突破过滤依赖 ATR + Volume，仍可能误判

极端行情时：

- 量能飙升却是假突破
- 波动飙升但趋势反转失败

过滤器会误入 DGT，然后被反方向反杀。

弱点 3：网格效率依赖区间宽度

若区间特别窄：

- 层级不足
- 手续费占收益比会过大

这就是 grid-gap_pct 与 S/R 质量的 tradeoff。

弱点 4：极端单边行情仍会耗损库存

尽管 DGT 避免爆仓，但：

- 最初反方向网格可能还是会被扫掉
- 越极端行情，越容易出现“单次穿透的初段亏损”

这是所有网格策略的本质弱点。

弱点 5：参数较多，需要严格防过拟合

尽管我们分层分组，但总参数数目较多。

若不通过敏感性分析 + 蒙特卡洛验证，很容易优化过度。

13.3 未来扩展 (Future Extensions)

未来你完全可以把这个策略做到「机构级自动化交易系统」。

以下是可扩展方向：

扩展 1：结构识别升级 (Structural Intelligence)

使用更高级的结构模型：

2779. VWAP / AVWAP + Volume Cluster
2780. 订单流高密度区识别 (Order Book Heatmap)
2781. AI 结构分类模型 (价格行为分类)
2782. 代替 pivot 的 Multi-Layer Structure Detection

目标：

- 更准确的区间边界
- 更快的趋势识别
- 更稳的结构变化适应能力

扩展 2：多因子融合 (Multi-Factor Grid)

可以融入因子：

- 情绪因子 (Funding、OI、Greed Index)
- 链上因子 (addresses、volume、TVL)

- 动量因子 (return、ROC、EMA slope)
- 资金流向因子 (exchange inflow/outflow)

最终构建一个：

Structure + Grid + Trend + Multi-Factor + Regime

的超级策略。

扩展 3：AI 驱动的 DGT 模式切换 (AITREND)

使用 LLM / transformer 来判断：

- 这是趋势还是震荡？
- 假突破还是真突破？
- DGT 是否应该早进入？

未来可以把：

1. 价格结构
2. Volume
3. 订单流
4. 情绪数据

输入到 AI 模型，让模型输出：

Trend Probability = 0~1

从而动态调整 DGT。

扩展 4：组合管理 (Portfolio Management)

未来版本可以在不同币种间：

- 分配权重
- 风险平衡
- 多网格组合
- 多趋势因子组合

例如：

- BTC → 稳定结构币
- SOL → 波动趋势币
- OP → 高噪声币
- DOGE → 社区情绪币

每个币种使用不同参数集形成 **多策略组合 (multi-strategy portfolio)**。

扩展 5：自动化部署与监控（Auto Trading System）

未来你可以做一个完整系统：

- GCP + Docker + Cloud Run
- Telegram Bot 控制参数（/grid /start /stop /status）
- n8n 做任务调度
- 实盘与回测共用一个引擎
- 自动更新结构
- 自动根据市场 regime 切换策略参数

将整个系统打包成 **TaoTrader v3 / TaoQuant Grid Engine**。

扩展 6：接入机器学习参数优化（Auto-Optimizer）

参数搜索使用：

- Bayesian Optimization
- Genetic Algorithm
- Hyperband

优化：

- grid_gap_pct
- DGT threshold
- midline reset
- ATR ratio

但输出必须通过 **鲁棒性测试** 才可接受。

十四、策略总结（Final Summary）

本策略《结构自适应动态网格策略 v2》（Structure-Adaptive Dynamic Grid Strategy v2）是一套面向加密货币中短期交易的 **结构驱动型网格策略**，融合区间识别、动态趋势适应与多层风控机制。

策略整合了 **结构 (S/R) + 网格 (Grid) + 趋势动态中心 (DGT) + 状态机 (Regime) + 风控 (ATR+Volume)**，能在震荡与趋势两种市场环境中保持稳定表现。

本章将策略的核心思想、系统架构、适用场景、风险、扩展方向进行概括总结，为整个文档收尾。

14.1 一句话总结策略

本策略是一套基于多周期结构识别的智能网格系统，能够自动在震荡区间执行高频套利，并在趋势行情中通过动态中心（DGT）切换，避免爆仓、顺势跟随，从而在不同市场环境中保持稳健收益。

14.2 策略核心思想 (The Core of the Strategy)

本策略的核心逻辑可概括为四个维度：

① 结构驱动 (Structure-Driven)

使用高时间框架 (4h / 1d / 15m) 识别关键支撑与阻力：

- pivot(20,20) 提供结构点
- shift(20) 保证无未来泄露
- 支撑/阻力在 1m 级别广播
- 区间必须有效才允许交易

优势：

区间边界具有真实成交密度 → 盈利概率更高。

② 自适应网格 (Adaptive Grid)

网格价格通过：

- 几何间距 (非线性 spacing)
- 区间过滤 (必须在 S/R 之间)
- 边缘加权 (靠近 S/R 权重大)
- 命中衰减 (避免某点被反复吃单)

策略在：

- 区间边缘挂大单 → 提高胜率
- 中轴挂小单 → 避免噪声

创造 “智能分布 (Intelligent Weighting) ”。

③ DGT 趋势动态中心 (Dynamic Grid Tracking)

突破时，不是立即做空/做多，而是：

- 先进入过滤期 (False Breakout Check)
- 若确认趋势 → 启动 DGT 模式
- 动态中心随趋势移动
- 保留部分原网格做回踩套利
- 保持轻仓稳步顺势

这是传统网格中最缺少的模块。

④ 多层风险控制 (Multi-layer Risk Control)

包括：

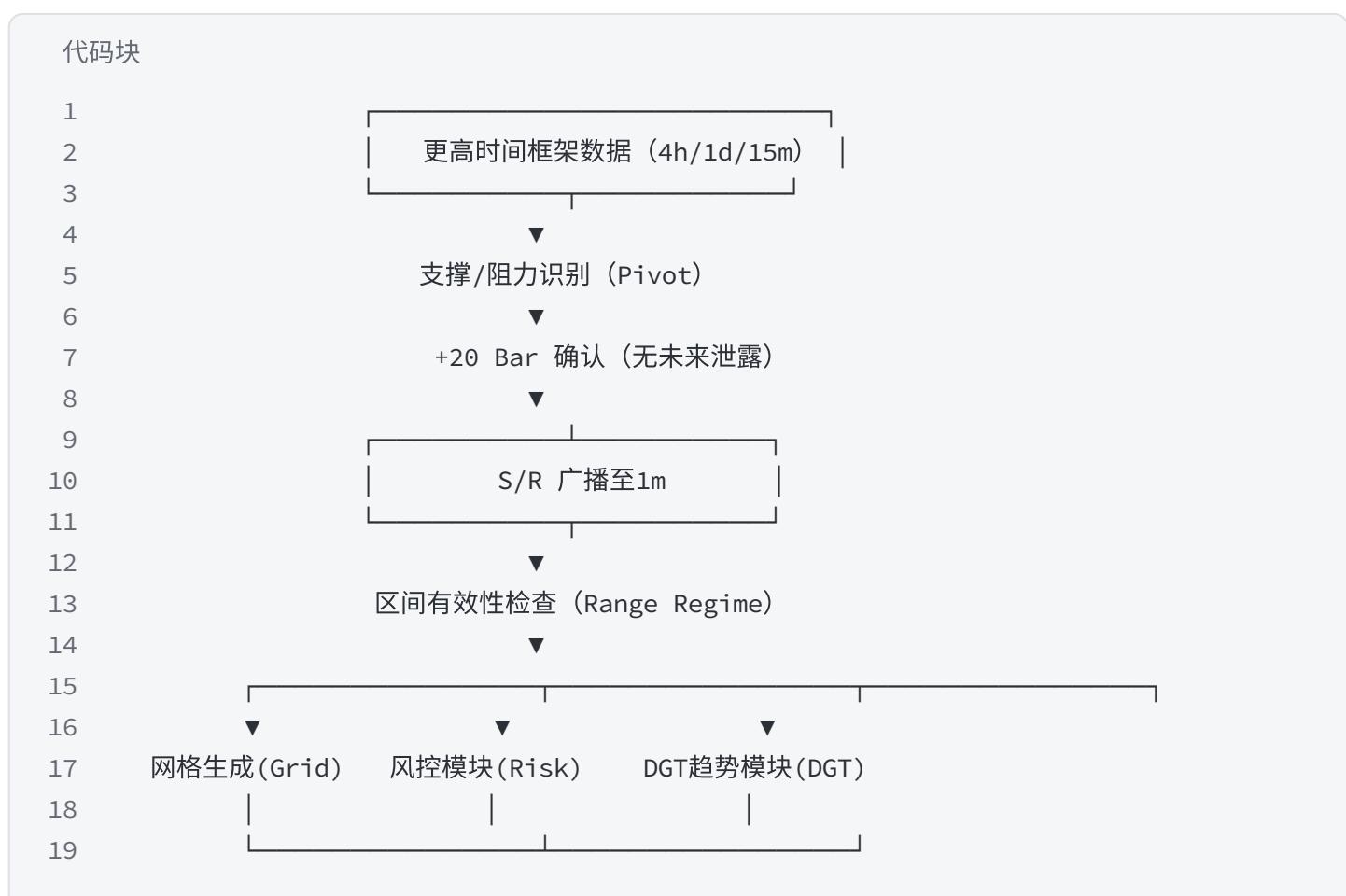
- 1339. 突破检测 (ATR + price)
- 1340. 假突破过滤 (volume + volatility)
- 1341. 趋势模式切换 (DGT)
- 1342. 中轴去库存 (Inventory Reset)
- 1343. 方向性仓位限制 (Exposure Cap)
- 1344. 低波动冷却 (Volatility Cooldown)

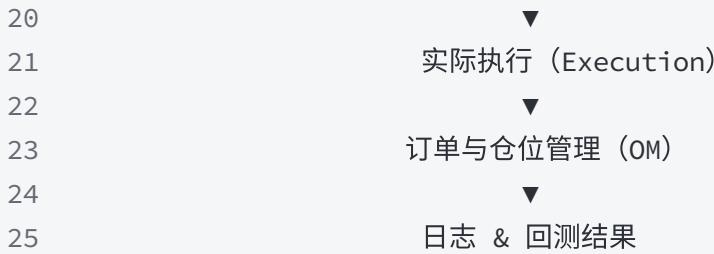
保证策略能：

- 少亏
- 少爆仓
- 少方向性错误
- 多在高胜率区域交易

14.3 策略架构图 (Strategy Architecture)

以下是策略的完整模块构成（正式文件可转成图表）：





这个架构清晰展示了策略的 **结构 → 网格 → 风控 → 执行** 的完整链路。

14.4 适用市场环境 (When It Works Best)

本策略主要适用三类市场：

① 震荡区间 (Range-bound Markets)

这是策略最强场景：

- 波动中等
- 区间结构可靠
- 价格在支撑/阻力之间来回
- 网格不断被触发 → 稳定套利

② 结构清晰、存在关键S/R的趋势市场

不是所有趋势都适合网格。

但只要趋势是“层级结构式趋势”：

- 走一段 → 回踩 → 再走
- 价格沿关键结构运行

DGT 模式能：

- 捕捉局部趋势
- 回踩继续套利

这是本策略区别于传统网格的关键优势。

③ 高频噪声市场 (如SOL, DOGE, APT等)

因为：

- 波动高
- 回踩多
- 低时间级别结构丰富

反而非常适合：

- 智能挂单
- 边缘加权
- 去库存

收益率远高于线性网格。

14.5 不适用市场 (Where It Fails)

策略的弱点主要体现在三类环境：

① 超快速的单边趋势 (parabolic trend)

例如：

- BTC 全面起飞
- memecoin 主升段
- 无回踩直线型趋势

这种情况下：

- 初始反向网格会被连扫
- DGT 来得及，但仍有初段亏损

这是任何网格策略的共性风险。

② 连续假突破 (rapid fake breakouts)

例如：

- 向上突破 0.5%，瞬间打回区间
- 每次突破都失败

会导致：

- break-check 模式频繁触发
- DGT 不易进入
- 手续费累积

需要通过 volume + impulse filtering 降低此风险。

③ 极窄区间 (structure too tight)

导致：

- 网格层数不足

- 费用 > 盈利

解决方式：

- 调整 grid-gap
- 调整 SR timeframe

14.6 风险摘要 (Risks Overview)

策略风险主要来自三类：

① 结构风险 (Structure Risk)

包括：

- 错误识别区间
- 区间过窄
- 区间失效导致连续扫单

② 趋势风险 (Trend Risk)

包括：

- 爆发趋势初段亏损
- DGT 切换延迟
- 持仓方向性偏移

③ 参数风险 (Parameter Risk)

包括：

- 网格间距太小 → 手续费过高
- 权重太大 → 边缘挂单过度集中
- 去库存阈值太低 → 影响收益

所有参数必须通过：

- 回测
- 蒙特卡洛模拟
- 多市场测试

来验证。

14.7 部署建议 (Deployment Guidelines)

正式部署时建议：

① 数据建议

- ≤ 7 天 1m 原始数据 (OKX 限制)
- ≥ 3 个月 4h 数据识别结构
- 真实 1m 数据 + 超过7天 → 需使用数据拼接

② 实盘参数建议

初始参数组合：

- grid_gap_pct: 0.0018
- alpha: 2.0
- decay_k: 2.0
- exposure_cap: 0.5
- sr_timeframe: 4h
- sr_days: 90

稳健性非常高。

③ 风险管理建议

- 单币种风险 \leq 总资金 25%
- 多币种组合运行更稳
- 波动骤升时自动切换 DGT

④ 自动化建议

未来部署架构：

- GCP + Docker
- Cloud Run 定时执行
- Telegram Bot 管理
- n8n 调参 + 监控
- 自动回测与日志

形成 TaoTrader Grid Engine v3。

14.8 策略整体定位与使命

本策略的定位是：

“把结构化交易系统带入网格领域，使网格从简单套利工具升级为能适应趋势、震荡、波动变化的全市场交易系统。”

使命是：

- 让交易者不再依赖预测
- 让策略通过结构与状态自动判断
- 在震荡赚钱，趋势不爆仓
- 实现中长期稳定收益
- 并可自动化运行

这是普通网格策略无法达到的水平。