arXiv:2611.30195v1 [cs.CV] 23 Nov 2026

面向动作的量化，面向部署的执行：世界动作模型的精度与执行联合选择

Tianqi Li[1], Xin He[2], Xue Geng[3],LeZhang[4],Ming-MingCheng[1,5,6],YunLiu[1,5,6,†]
VCIP，南开大学计算机科学与工程学院[2]天津理工大学计算机科学与工程学院[3]新加坡科技研究局资讯通信研究院[4]电子科技大学信息与通信工程学院[5]南开大学前沿交叉科学研究院[6]南开国际先进研究院，深圳福田

†通讯作者。

摘要

世界动作模型通过预测动作的效果或根据观测生成指令来指导机器人。模型在执行动作前可能会比较数百个候选未来，并反复读取必须装入设备内存的权重。训练后量化减少了存储预训练权重所需的位数，但更小的模型不一定可部署：临时张量可能超过图形处理器（*GPU*）的内存限制，而即使很小的预测误差也可能改变所选动作。这带来了两个相互耦合的挑战：确定能够保持任务性能的位宽分配，并确定其中哪些满足实际执行约束，同时避免耗尽昂贵的任务试验。我们提出***PTQ4WAM***，这是一个在内存、延迟和任务评估预算约束下选择逐层位宽分配与执行路径的框架。校准测量用于指导候选方案提议，也可以选择使用语言模型智能体。在为可行分配方案投入任务试验之前，我们先在图形处理器上对全模型执行路径进行性能分析。融合分组重缩放并原地读取视觉历史，可以使原本不可行的分配方案变得可部署。在来自四个模型系列的*15*个工作负载中，中位数任务得分达到相应*32*位浮点（*FP32*）得分的***99.2%***。在固定的*GPU*重放中，相对于*FP32*的中位数加速比为 **2.36**×，峰值分配内存的中位数缩减为*71.5%*。代码即将发布。

1. 引言

机器人在选择下一步动作时，可以预测若干动作会将自己带到哪里，然后执行最接近目标的那个动作。DINO-WM 和 LeWorldModel（LeWM）在学习得到的视觉特征空间中进行这些预测 [19, 33]。V-JEPA 2-AC 使用基于预训练视频特征构建的预测器进行规划 [3]。Fast-WAM 使用视频与动作监督直接生成命令 [31]。这些世界动作模型（WAM）必须与其处理的观测和中间特征一同适配运行。预测式规划器还会反复使用相同的权重。在一种实体机器人设置中，V-JEPA 2-AC 使用一个含 3 亿个参数的预测器，并在每次十轮迭代优化中比较 800 个动作候选项。选择一个动作需要在 RTX 4090 [3] 上耗时 16 秒。机载控制器必须满足内存预算，同时还要在截止时间前选出下一个动作。

训练后量化（PTQ）提供了一种缩小预训练 WAM 的方法，无需收集新的机器人交互数据进行再训练。每个权重使用更少的位，可以减小保存的模型以及算子每次调用时必须读取的数据量。在固定的存储预算下，某些层保留更多位，就意味着其余层可用的位更少。只有少量任务评估时，我们如何找到一种配置，既能保持控制性能，又能在设备限制内运行？

任务对齐。 考虑两个候选轨迹，其预测端点与目标的距离几乎相同。较小的特征误差就可能改变它们的排序，进而改变动作。而将两个预测结果一同移动的较大分布偏移，反而可能不会改变决策。因此，平均重构误差并不能可靠地指示哪些层需要更多位。近期的 WAM 已经利用动作敏感性和滚动反馈来指导量化 [1, 27, 34]。若要选择整个模型，我们需要了解其精度选择在原始规划器或控制器执行动作时如何协同发挥作用。

反馈成本。用于校准的代表性输入可以在一次遍历中揭示许多层中的数值误差。要测试某个完整分配是否能达到控制成功率，可能需要从许多初始状态启动控制器并运行，直至每次试验结束。每个动作都可能改变观测

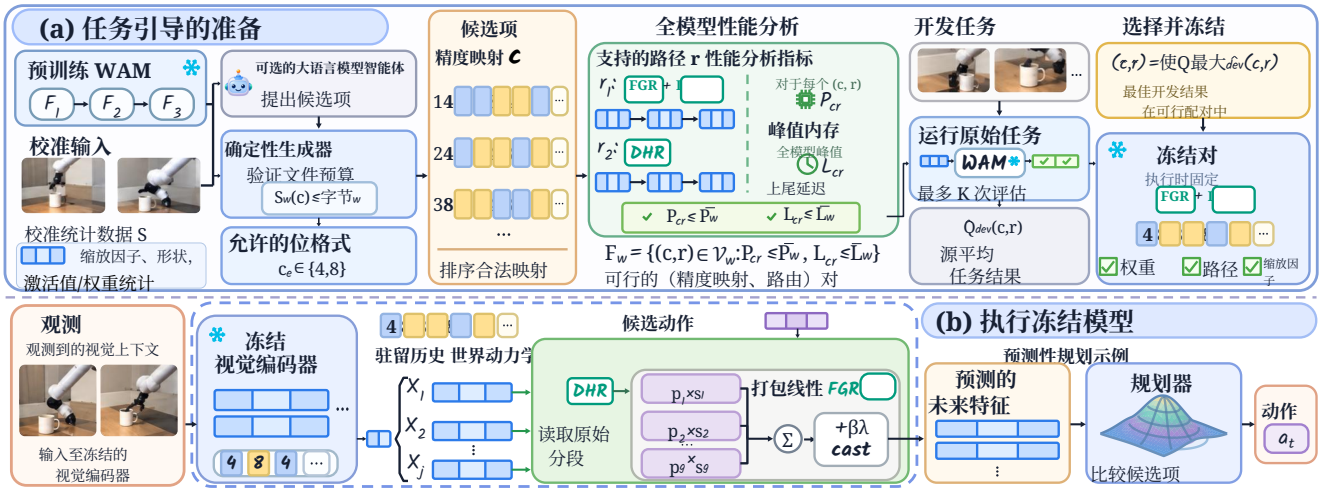

图1. **精度选择与部署**。 (a) PTQ4WAM 对精度映射和执行路径进行性能分析，然后根据任务得分选择可行的组合。(b) 规划器比较冻结模型预测的未来，以选择一个动作。

，供下一次决策使用。测试所有位数分配方式会耗费大部分准备时间来运行任务。在预算有限的情况下，我们必须仔细决定哪些分配真正值得在任务上运行。

运行时可行性。 相同保存大小的位宽分配，在预测过程中是否能运行可能有所不同。量化层可能会在重新缩放之前写入临时结果。对历史观测的注意力计算可能会先将历史记录复制到新的张量中。峰值取决于这些张量中哪些与模型的其余部分同时存在。单个内核的速度或权重文件的大小，无法告诉我们完整调用能否在内存和时间限制内完成。在不可行模型上花费一次试验，就少了一次测试设备能够运行的模型的机会。

PTQ4WAM 首先测量设备能够运行哪些位宽分配，然后利用有限的任务评估，选出性能最佳的模型及其图形处理器实现。例如，原位读取历史记录，可以使超出内存限制（若复制历史记录）的模型变得可行。随后，该映射便可纳入图1(a) 中的任务比较。图1(b) 展示了规划器如何使用所选模型。

- **设备约束的精度选择**。 我们使用原始任务得分来比较全模型比特分配。每种分配都会经过测试，并与满足设备内存和时间限制的 GPU 内核一同保存。
- **校准引导的提案**。 我们可以使用一个具备模型结构和校准测量值的大语言模型（LLM）智能体，提出一小组分配方案。确定性生成器会先检查这些方案，然后规划器或控制器再花时间为其评分。
- **低内存执行**。 我们在低比特算子内部重新缩放输出，并直接从现有缓冲区读取视觉特征。我们测量完整调用的耗时和内存占用，以找出同时满足两项限制的路径。

2. 相关工作

如何选择位宽。 先前工作会校准区块，利用曲率选择权重或位宽，并调整词元分组的激活值范围 [7, 8, 15, 16]。AdaRound 学习如何舍入权重，QDrop 改变被量化的激活值，Q-BERT 和 PTQ4ViT 则使用曲率信息 [20, 24, 28, 32]。任务损失或模型的敏感性可用于指导应保护模型的哪些部分 [1, 21, 25, 27, 30]。QuantWAMs 利用滚动反馈保护选定的去噪步骤 [34]。多个组件产生的误差可能改变规划器的决策 [10]。因此，局部评分可能会错误地排列两个完整模型的优劣。对于每种映射，我们都会在其满足两项设备限制后，在任务上进行一次测试。

如何使用校准数据。 校准可以通过重新缩放或学习得到的变换来重新分配离群值 [17,23, 29], 旋转激活值 [2, 18], ，也可以在低秩分支中保留离群值 [14]。设备测量值可以指导比特分配 [26]。在约束贝叶斯优化（BO）中，先前试验的结果会用于指导下一步选择 [11]。在确认成功之前，单次试验可能需要进行大量预测。我们采用的可选 LLM 会利用校准数据，在首次试验前提出不同的映射。我们只将可用试验用于满足设备限制的映射。

如何减少内存使用。 张量编译器和内核库会融合操作或对矩阵乘积进行分块 [4, 5,22]。Marlin 和 FLUTE 可降低低位宽矩阵乘积的开销 [9, 12]。注意力内核可减少临时存储空间，或读取独立的内存区块 [6, 13]。如果峰值由另一个算子决定，那么移除一个缓冲区对模型层面的内存使用帮助不大。我们会对每次完整调用进行性能分析，测试融合重缩放或直接历史读取，以确定哪些映射能同时满足两项限制。

3. 方法

我们从一个预训练 WAM 及其原始规划器或控制器开始。任务会规定机器人应执行的操作以及评分方式。例如，当物体到达目标位置时，推送任务即告成功。工作负载 w 会为该任务指定模型、输入形状、历史长度和规划时域。精度映射 c 为每个符合条件的组件分配权重位宽。执行路径 r 则指定用于运行该映射的图形处理器操作。每条路径都使用相同的存储权重和量化参数执行。

校准输入用于设定量化尺度、指导候选提案，并提供分析所需的形状和特征。我们根据开发结果在模型中进行选择，并在留出数据上评估该选择。每个源都是一个规划实例、一条记录的轨迹，或一次从初始状态运行至终止的控制器运行。我们将整个源（包括其中的每一帧和结果）分配到同一个数据划分中。

3.1. 设备约束的精度选择

匹配存储模型大小。对于工作负载 w，决策包括精度映射 c 和执行路径 r。组件 i包含 n_i 个权重，每个权重使用 b_i 位，即四位或八位。令 $S_w^{\text{aux}}(c)$ 表示辅助字节数。将该开销加到权重载荷上，即得文件总大小：

$$S_w(c) = \sum_i \lceil n_i b_i / 8 \rceil + S_w^{\text{aux}}(c). \tag{1}$$

辅助字节数包括缩放因子、零点、受保护的组件、填充和头部。在字节数有硬性限制的部署中，B_w 必须满足 $S_w(c) \le B_w$，如图1(a) 所示。对于大小匹配实验，我们则允许与目标值存在绝对相对偏差 10^{-3}，并在比较中对所有映射采用相同的激活精度。

识别可行实现。令 $\mathcal{V}_w$ 包含满足存储规则且通过校准输入数值检查的受支持配对。我们使用部署形状和历史记录对每个配对进行性能分析。令 P_{cr} 表示进程峰值内存，L_{cr} 表示上尾延迟。联合搜索采用完整调用延迟的第 95 百分位数（第 95 百分位）。$\bar{}$ $\bar{}$设备设定内存限制 P_w 和延迟限制 L_w。我们将同时满足这两项限制的配对集合定义为

$$\mathcal{F}_w = \{(c, r) \in \mathcal{V}_w : P_{cr} \le \bar{P}_w,\ L_{cr} \le \bar{L}_w\}. \tag{2}$$

进程内存包括运行时及其库所持有的分配。对于每个具有可行路径的映射 c，我们在观察任何任务结果之前，将其最快的此类路径固定为 r_c。直接历史读取可能使某个映射符合条件，而该映射原本会超出上限。如果所有受支持的配对都至少违反一项限制，则该设备没有可行候选项。

按任务结果进行选择。令 $\mathcal{E}_K \subseteq \mathcal{F}_w$ 为获得 K 种可用任务评估之一的候选对。对于在开发源 $Z_1, \dots, Z_n$上测试的一对，令$v(c, r, Z_j) \in [0, 1]$ 表示其在源内的平均结果 j。开发集得分对每个源赋予相同的权重，

$$\widehat{Q}_{\text{dev}}(c, r) = \frac{100}{n} \sum_{j=1}^{n} v(c, r, Z_j). \tag{3}$$

对于对照组，每个源都从指定的初始状态开始，并记录执行是否达到目标。对于记录数据，一个源可以贡献多项比较或多个时间窗口。在源内取平均值可避免较长的录制数据在模型选择中占据过大权重。部署对为

$$(\hat{c}, \hat{r}) \in \arg\max_{(c,r) \in \mathcal{E}_K} \widehat{Q}_{\text{dev}}(c, r). \tag{4}$$

若出现并列，依次按延迟、进程内存和候选顺序决定。除非已知某条替代路径在整个任务计算过程中会产生完全相同的输出，否则采用该路径需要额外进行一次评估。部署时，使用保存的路径以及选定的权重和对应的缩放因子。

3.2. 校准引导的提案

使用方程（3）测试一种分配方案，所需的不仅仅是一次前向传播。在对照组中，第一个动作决定下一个观测，因此任务评估器必须跟踪由此产生的轨迹。我们利用校准期间可获得的信息，判断哪些映射值得付出这项开销。

构建一个合法映射候选池。交换大小相同的组件的位宽，可以在模型内重新分配精度，同时保持权重位数总量不变。在联合搜索实验中，我们从若干存储目标相同的映射开始。交换次数越多，精度就离起始映射越远。我们会去除重复项、核验每个文件的大小，并向每种策略提供相同的候选池。

优先考虑彼此不同的映射。可选的大语言模型会接收每个组件的形状、与其他层的连接关系、校准统计数据、允许的格式和存储预算。它可以为某个组件保留八位，或限制采用四位的组件数量。确定性生成器会强制执行这些约束，并检查打包文件。为保持多样性，它还要求映射之间的汉明距离d_{H} 达到最小值 $d_{\min}$。我们用 $\mathbf{1}[\cdot]$表示一个指示函数：当其条件成立时取一，否则取零。对于列表中的任意两个映射，我们要求

$$d_{\text{H}}(c, c') = \sum_i \mathbf{1}[b_i(c) \ne b_i(c')] \ \ge d_{\min}. \tag{5}$$

这里，$b_i(c)$表示映射 c 分配给组件 i的位宽。在首次计算开发集得分之前，我们会修复无效提案并固定顺序。在联合搜索中，所有策略都会先查看相同的路径配置文件，然后再选择映射。

候选名单可能会排除最佳映射，而有限的开发样本也可能导致名单中的错误映射胜出。考虑在评估前固定的 K 候选项，其效用取自 $[0, 1]$ 范围内的 n 独立源。对于失败概率 $\delta \in (0,1)$，经验最优者的期望效用以至少 $1-\delta$ 的概率落在候选池中最佳效用的 $(2 \log(2K/\delta)/n)^{1/2}$ 以内。将此界乘以 100，即可得到缩放因子上的百分比

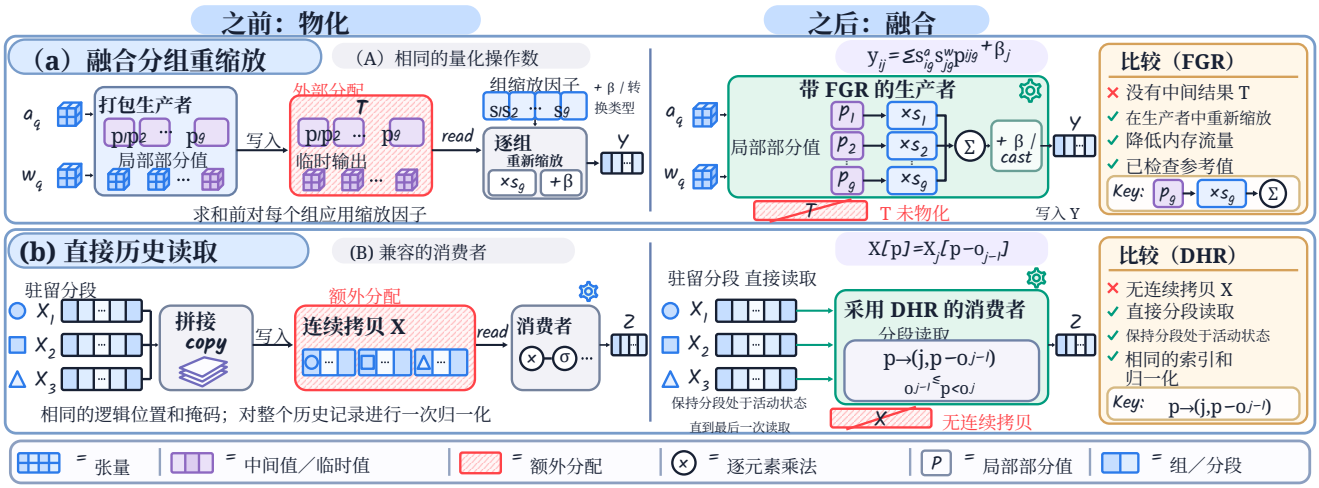

图2. **同一组量化运算的两种路径。** (a) 融合分组重缩放在写入输出之前应用每个缩放因子。(b) 直接历史读取会保留原始张量，直到消费者完成操作。

Equation (3) 的缩放因子上的百分点。候选项可以共享源。更多源可减少采样误差，但要找回被遗漏的映射，则需要建立新的候选池。

在初始状态之间分配试验次数。我们在相同的初始源上测试每个入围映射，并仅扩展表现更好的映射。令 S 表示阶段数。在阶段 s，剩余的 K_s 个映射各自已在 n_s 个源上进行过测试。对于 $n_0 = 0$，我们将新增的候选项—源访问总次数计为

$$N_{\text{eval}} = \sum_{s=1}^{S} K_s(n_s - n_{s-1}). \tag{6}$$

每个新增源仍会运行至其通常的停止点。此安排在保留每个源的规划时域的同时，节省了完整试验的次数。校准、性能分析、打包和编译也会计入总准备时间。

3.3. 低内存执行

融合分组重缩放 (FGR) 和直接历史数据读取 (DHR) 可保留任务输出。图2(a) 描绘了线性层中的部分和，而图2(b) 展示了消费者读取过去和当前特征的位置。

融合分组重缩放。量化线性层会针对每组 g 输入通道、输出行 i 和输出通道 j，分别计算一个整数部分点积 p_{ijg}。单独的重缩放算子需要将这些中间值写入设备内存再读回。FGR 会在计算每个部分值时对其重新缩放。给定激活缩放因子 s^a_{ig}、权重缩放因子 s^w_{jg} 和偏置 β_j，该算子会为该行和输出通道写入单个值 y_{ij}，其计算方式为

$$y_{ij} = \sum_{g} s^a_{ig} s^w_{jg}\, p_{ijg} + \beta_j. \tag{7}$$

不同分组的缩放因子可能不同，因此需要先重新缩放再求和。内核会添加偏置，并在最终写入前将输出转换为规定格式。

此计算的参考值是使用相同量化码、分组和零点的线性层。每个整数部分结果都必须能由其累加器表示。如果满足这些条件，将缩放乘法移入生产者，就能在实数运算中保留该算子。改变浮点加法的顺序仍可能导致舍入后的输出发生变化。因此，在融合路径进入搜索之前，我们会检查它是否与量化参考值一致。

直接历史读取。预测器可以将历史特征和最新观测保存在独立的张量中。令$\mathbf{X}$ 表示它们的逻辑拼接。分段$\mathbf{X}_j$ 在独占偏移量 o_j 处结束，其中 $o_0 = 0$。DHR 不会将这些分段复制到新的缓冲区，而是找到包含位置 p 的分段，并从该分段中读取元素，

$$\mathbf{X}[p] = \mathbf{X}_j[p - o_{j-1}], \quad o_{j-1} \le p < o_j. \tag{8}$$

注意力会保留原始位置和掩码，并对整个历史记录进行一次归一化。分别对每个分段进行归一化会改变过去和当前观测的相对权重。只要两个分段保持不变，且消费者使用其值和逻辑位置，直接读取就是有效的。如果消费者会通过输入进行写入，或要求连续地址范围，则仍需复制。在消费者完成读取之前，两个分段都必须保留在内存中。

检查完整预测过程的峰值。 FGR 可能需要工作区，而 DHR 会让输入分段一直保持有效，直到最后一次使用。令 A 表示变更阶段的原始分配，d_{save} 表示该阶段净节省的字节数，C 表示任何未变更阶段的最大分配。此变更会使峰值降低以下字节数，

$$\Delta M_{\text{peak}} = \max(A, C) - \max(A - d_{\text{save}}, C). \tag{9}$$

当另一个阶段决定峰值时，即使峰值保持不变，在发生变化的阶段移除更多字节仍可节省流量。因此，我们会测量每个

路径，然后应用方程（2）中的限制。

4. 实验

表 1 涵盖 15 个工作负载及其原始规划器或动作接口和预处理方式。前 13 行的质量指标为规划或控制试验的成功比例。对于 Franka Ranking，我们统计模型将两条记录的动作序列按正确顺序排列的频率。对于 Robo- Casa Windows，我们统计结果为正的轨迹窗口所占比例。对于每个工作负载，我们将量化后的分数除以其原始 FP32 检查点的分数。十五个工作负载的权重相同。

对于记录任务而言，观测单位最为重要。Windows 部署面板包含 14 条源轨迹，每条轨迹贡献四个短窗口。正向窗口表示该分段的结果为正。若要在整条轨迹上衡量成功，则需要轨迹级结果。对于排序任务，正确比较意味着模型在记录的配对中更偏好参考动作序列。这两项分数均来自记录数据。要对机器人本身计时，就必须在物理系统上运行这些动作。

即使模型变慢，较早达到目标的机器人也可以减少预测步数。因此，对于表1，我们在加载并完成编译后，于 RTX 4090 上对相同输入执行完全相同的 GPU 调用并进行重放。加速比为 FP32 重放时间除以 PTQ4WAM 重放时间。峰值内存是重放期间分配量的最大总和，包括临时张量，单位为吉比字节（GiB）。进程限制使用十进制兆字节（MB），阶段分配量使用米比字节（MiB），流量使用十进制吉字节（GB）。在固定时域下对完整的模拟器批次计时，可以显示重放所节省的时间有多少能够延续到控制中。

对于部署决策，我们测量整个进程的峰值内存。这包括分配器和 GPU 库预留的内存，以及模型使用的张量。六个工作负载在其存储预算下允许采用单一比特分配。这些案例衡量的是在权重格式固定时量化所节省的空间。选择实验使用可在多种分配之间进行选择的工作负载。补充材料提供了更多基线和分布偏移、配对测量、数值检查与证明，并记录了每项比较所使用的源数据划分。

4.1. 跨 15 个工作负载的部署

在固定输入上，四个系列的运行速度都更快，并且使用更少的图形处理器内存（表 1）。在大多数工作负载上，任务得分仍接近 FP32。对于 Franka Ranking，得分下降相当于多出现一次错误比较。Fast-WAM Long 和 LeWM Wall 都受益于更低的预测成本，但它们的对照组得分却朝相反方向变化。接下来，我们将考察每种精度分配保留了哪些动作。

从预测时间到控制时间。在 Fast-WAM Long 上，我们测量了固定时域下 500 回合的完整批次其中包括环境更新、调度和输入/输出。与优化后的脑浮点 16（BF16）相比，PTQ4WAM 将平均批次时间从 973.0 秒缩短至 675.7 秒，同时观测成功率相近（95.10 % 和 95.32%）。准备 PTQ4WAM 耗时 3,469.4 秒，而 BF16 仅需 48.6 秒。按测得的批次时间计算，初始成本在第十二个批次时即可收回。

部署比较同时改变权重和执行方式。我们固定候选池以比较评分规则，并固定试验预算以比较提案。在每条路径上运行相同的权重，可以隔离执行成本。

4.2. 设备约束的精度选择

哪些预测误差会改变动作？ 在 LeWM Wall 中，规划器根据预测后果对候选动作进行排序。即使预测特征几乎没有变化，只要扰动颠倒了排名靠前的动作顺序，机器人的路径就可能改变。我们通过固定规划器和初始状态，并用幅度匹配的随机噪声替代量化误差，来检验这一解释。我们测量标准化特征的均方误差。我们将第一个动作与来自更高预算规划或环境标签的参考动作进行比较，然后运行对照组直至终止。表 2 给出了恢复参考动作的比例以及试验成功的比例。

匹配噪声和 PTQ4WAM 报告的平均特征误差相同。选定映射与参考动作的一致性更高，成功率也更高。将校准判定为最敏感的组件恢复到更高精度后，特征虽更接近其参考值，但动作一致性和控制成功率均有所下降。随机恢复可控制恢复哪些层这一选择的影响。特征误差最低的映射未必能保留规划器原本会选择的动作。

Wall 在部署比较中相较于 FP32 的增益最大。为检验额外的调优试验是否能解释这一增益，我们为 FP32 规划器提供了 12 种开发配置，为 PTQ4WAM 提供了 12 项分配。在相同的留出状态上，量化规划器仍领先 11.07 个百分点。这一增益仅适用于该检查点和任务。在其他部署中，量化可能会保留、提高或降低整体任务得分。

我们还在保留检查点的同时，改变了 Wall 上的初始状态、目标和规划时域。在三项比较中，量化规划器的成功率都更高，严重失败也更少。当时域为通常时域的 1.5 倍时，严重失败率从 11.6% 降至 5.2%。此计数包括碰撞、不安全终止状态和超时。结合匹配噪声对照，这一结果表明，增益与滚动执行过程中做出的决策有关。此次比较中有多层同时发生变化，因此我们无法将增益归因于任何单个权重的舍入。

改变选择规则。表3 固定量化器、

表 1。 **任务质量与部署成本 涵盖十五个工作负载。** 时间和峰值分配是在相同的图形处理器重放 s.

模型	Task	质量（%）↑ FP32	质量（%）↑ Ours	时间（秒）↓ FP32	时间（秒）↓ Ours	峰值（GiB）↓ FP32	峰值（GiB）↓ Ours	加速比 (×)
Fast-WAM	空间	98.0	96.4	143.45	33.40	23.256	5.103	4.30
	物体	99.6	99.4	149.43	38.64	23.256	5.103	3.87
	Goal	97.2	96.4	145.04	33.95	23.256	5.117	4.27
	Long	95.4	93.4	13.17	6.57	23.151	6.608	2.00
LeWM	Wall	78.0	94.0	23.13	10.71	0.439	0.162	2.16
	Push-T	91.3	90.0	10.69	3.32	0.439	0.155	3.22
	TwoRoom	87.0	86.0	23.20	10.74	0.439	0.162	2.16
	Cube	78.0	84.0	11.86	4.17	0.440	0.153	2.85
	Reacher	94.0	92.0	11.34	3.07	0.439	0.133	3.70
DINO-WM	Wall	94.0	98.0	10.35	5.03	4.126	0.811	2.06
	PointMaze	100.0	100.0	14.16	5.99	18.254	0.857	2.36
	Push-T	92.0	92.0	41.24	19.92	18.254	0.863	2.07
	Meta-World Reach	90.0	88.0	138.76	58.91	2.460	0.848	2.36
V-JEPA 2-AC	Franka Ranking	93.3	91.1	37.51	12.24	5.370	1.415	3.07
	RoboCasa Windows	35.7	37.5	114.99	53.73	6.022	2.049	2.14

表2. **LeWM Wall 上的预测误差、动作一致性和控制成功率。** 我们在用于评估规划器的标准化特征空间中测量均方预测误差。

模型或干预	预测误差	动作平均值（%）	Task 成功率（%）
FP32	0.0217	80.6	78.03
匹配噪声	0.0225	81.9	79.87
恢复敏感项	0.0219	85.1	82.86
随机恢复	0.0223	90.8	90.25
置换精度映射	0.0231	84.6	83.74
PTQ4WAM	0.0225	**93.4**	**94.25**

表3。 **共享候选池上的留出任务得分（%）。**

工作负载	重建。	海森矩阵	Task
Fast-WAM Long	89.84	90.57	**93.67**
LeWM Wall	90.55	91.35	**94.36**
LeWM Push-T	86.79	87.52	**90.37**
LeWM TwoRoom	82.61	83.55	**86.18**
LeWM Cube	80.47	81.36	**84.01**
DINO-WM Push-T	88.66	89.55	**92.29**
Franka Ranking	87.63	88.32	**91.46**
RoboCasa Windows	34.09	34.79	**37.59**

字节预算、执行路径和候选映射，仅改变对它们的排序方式。重构用于衡量与原始模型输出的一致性。曲率加权会提高那些预计会改变其损失的误差的权重。按原始任务得分排序，可在每种工作负载下获得最高的留出得分，包括两个根据录制数据评分的任务。

我们还将 SteerQuant、QuantWAMs 和 Q- WAM [1, 27, 34]调整为相同的支持格式、存储目标和规划器，并采用 12 次任务评估的预算。表现最强的调整方法 Quant- WAMs 利用滚动反馈来保护去噪步骤。对完整赋值进行评分后，PTQ4WAM 在 Fast-WAM Long 上领先 2.21 个百分点，在目标上领先 1.45 个百分点。这些比较针对的是经过匹配调整的方法。在前述对照组中，这些规则对同一组权重进行排序。

何时检查设备限制。我们针对每个模型族的一个工作负载运行了 12 次独立搜索。每次搜索使用 64 个校准源、128 个开发源和 256 个留出源。所有策略都采用相同的 48 项分配、每项分配两条经过分析的路径，以及相同的进程内存和单次调用延迟上限。表 4 中的延后检查策略，按校准排序对 12 个映射评分后，再检查首选路径。PTQ4WAM 使用相同的排序，但将 12 次试验用于可行配对。BO 也获得 12 次试验。全池参考值（全部）则对 48 个映射各进行一次试验。

这种排序会带来前期成本。每次搜索都会为 48 个映射分别分析两条路径，每条路径进行 20 次预热和 100 次计时模型调用，因此在任务评分前共需 11,520 次调用。校准、调优、打包和失败尝试都计入准备成本。相比之下，在 Long 上进行一次任务试验，会在 128 个开发源上评估一个映射，并对每个源执行三次控制器执行。模型调用可以用来测试候选项是否适配设备，而无需承担完成这些完整轨迹的成本。为每种策略提供这些分析结果，可以隔离其任务试验的分配选择。

在选择前检查限制条件，提高了四种工作负载上的留出集得分。在试验预算固定的情况下，每个不可行的映射都会挤掉一个本可用于执行任务的候选项。延迟检查获胜者都通过了校准输入上的检查。最后一列统计了它们在多少次情况下也能在两个限制条件内处理所有留出源。其他三种策略在每种工作负载的全部 12 次搜索中，均满足这两个限制条件。留出源上的违规会触发

表4。 **留出集得分（%）及 12 次搜索中的可行性。**

	12 次试验			48 次试验	Late
工作负载	Late	BO	Ours	All	符合/12
Fast-WAM Long	93.38	93.13	**95.17**	95.24	8
LeWM Push-T	89.76	88.77	**91.56**	91.70	9
DINO-WM Push-T	91.88	91.62	**93.64**	93.75	10
RoboCasa Windows	37.52	37.15	**39.95**	40.08	9

一次失败尝试，并使用固定回退方案重新运行。报告的任务得分包含回退方案的结果。

贝叶斯优化利用较早的任务得分来选择下一个可行候选项。PTQ4WAM 仅根据校准测量值对相同的候选项排序。在相同的 12 次试验预算下，其得分在每项工作负载上都更接近全池参考值。该参考值衡量了所有搜索策略均见过的 48 个分配中可达到的质量。

4.3. 校准引导的提案

哪些候选项值得进行试验? 我们固定了任务评分规则，并改变候选名单。在六项权重相同的规划与控制任务中，智能体使用 12 个候选项时的平均成功率达到 90.93%，接近对全部 48 个候选项进行评分时的 91.00%。在相同的 12 个候选项预算下，表现最强的替代方案——敏感性引导列表——达到 90.37%。随机列表和固定列表的得分均低于敏感性引导列表。

测试更多候选项是否总能找到更好的模型？在另一项 Fast-WAM Long 实验中，我们在收集任何任务得分之前冻结了嵌套候选池，并为每种候选池规模使用相同的 384 个开发结果。从 48 个候选项扩展到 192 个，获胜的开发计数从 368 增至 373。留出成功率分别为 94.36% 和 94.33%，而准备时间则从约 272 秒增至 1,017 秒。在这项实验中，较大的候选池提供了更多拟合开发样本的机会。在开发源上看似取得的增益，在我们将选定模型用于新源时便消失了。

哪些信息用于指导候选短名单? 在 Fast-WAM Long 上，我们打乱了形状兼容的组件之间的校准统计数据。12 项分配的值、模型结构和预算保持不变。省略这些统计数据，可以检验它们对智能体的价值。图3还将智能体与使用正确统计数据的确定性规则进行了比较。

向智能体提供生成各层测量值的层信息，得到了最佳候选短名单。在各层之间打乱这些值，效果不如完全不提供这些信息。在使用相同测量值的情况下，确定性启发式方法的表现也逊于智能体。我们通过在留出数据源上运行原始控制器，对全部四个候选短名单进行了评估。

提出并测试候选项的成本。 在八种受控工作负载中，任务评估的准备成本占主导。构建智能体候选名单的成本略高于应用固定规则。其主要节省来自它避免的

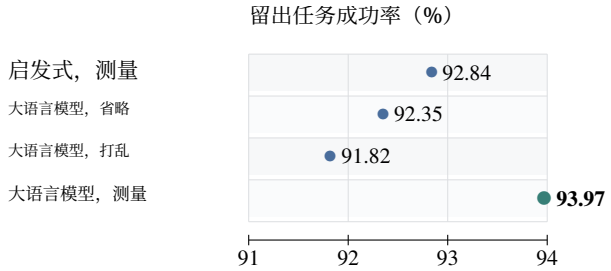

图3. **Fast-WAM Long 上的校准输入与提议规则。**每个候选名单都包含 12 个存储预算相同的映射。

试验。每张映射的初始状态数量是另一个节省来源。在 Fast-WAM Long 上，分阶段评估将候选项-源访问次数从 1,536 次降至 512 次，同时保留了每次试验的完整时域。分阶段搜索的留出成功率达到 94.94%，而所有映射使用每个开发源时为 95.17%。

在分阶段比较中，全部 12 个候选项都从相同的 16 个开发源开始。4 个候选项继续使用 64 个源，另有 2 个继续使用 128 个源；两个阶段均复用先前的结果。平均准备时间从 8,275 秒降至 3,496 秒，其中包括路径分析和编译。全部 12 次分阶段运行都在 4,000 秒的准备时限内完成。留出成功率的差异衡量了在 Long 工作负载上提前淘汰候选项所付出的代价。

4.4. 低内存执行

固定精度。针对每个模型族中的一个工作负载，我们在相同输入上运行三种实现，并固定权重、缩放因子和位宽（表5）。基线读取打包权重，并分别运行每项操作。经过调优的低比特实现获得与 PTQ4WAM 相同的校准和调优额度。在每个评估源上，这三种路径的任务结果都相同。因此，测得的时间差异反映了这些固定权重所采用的执行路径。

在第4.2节的 LeWM Wall 地图上，分别应用缩放因子会得到相同的参考动作一致性和控制成功率。两种路径都保留了选定映射的决策。

我们还在原生 Marlin 支持的一种格式下进行测试：每个权重使用 4 位，每个激活使用 16 位 [9]。为 Marlin 添加分段历史读取器可以缩短重放时间，而完整的 PTQ4 WAM路径则进一步加快了速度。这个中间对照组让我们能够区分组装历史记录的开销与低比特线性层所耗费的时间。

对一个固定模型步骤的跟踪记录展示了部分时间的去向。编译适配器执行了 184 个内核，与设备内存之间传输了 1.284 GB 数据。PTQ4WAM路径执行了 151 个内核，传输了 1.036 GB 数据。FGR 应用于 40 个符合条件的位置中的 36 个，DHR 应用于 9 个中的 7 个。其余位置使用了回退实现。跟踪记录包含发生回退的位置，因此

表 5. **冻结精度映射的重放时间和峰值分配**。计时涵盖在同一图形处理器上执行的 100 次调用。我们以兆字节为单位给出峰值分配内存，并将每个数值四舍五入到最接近的整数。

	重放时间（秒）↓				峰值分配内存（MB）↓			
工作负载	BF16-opt	打包	调优后	PTQ4WAM	都不是	FGR	DHR	Both
Fast-WAM Long	10.326	8.071	7.337	**6.352**	7681	7103	7390	**6846**
LeWM Push-T	5.082	3.958	3.611	**3.199**	197	174	185	**161**
DINO-WM Push-T	28.640	22.170	20.918	**18.660**	1723	1327	1218	**936**
RoboCasa Windows	80.420	61.910	57.545	**51.118**	2810	2409	2538	**2206**

流量的缩减并不要求每个位置都进行融合。

哪个操作决定了内存峰值？ FGR 移除了计算组输出与重新缩放之间的写入操作。使用 DHR 时，算子读取历史特征而不进行复制。每次只启用一项更改，便能揭示它们各自不同的影响（表5）。在 Fast-WAM Long 上，重新缩放节省的峰值内存更多；而在 DINO-WM Push-T 上，读取历史数据节省的更多。两项更改同时启用时，四个案例的峰值都最小。其效果取决于一次预测调用期间同时存在的中间张量。

图4更详细地追踪了历史副本。原地读取分段可节省该阶段的存储空间，但消费者可能需要额外的工作区。一旦另一个阶段占用了最大分配，在原阶段进一步节省也不会改变模型峰值，如公式（9）所示。

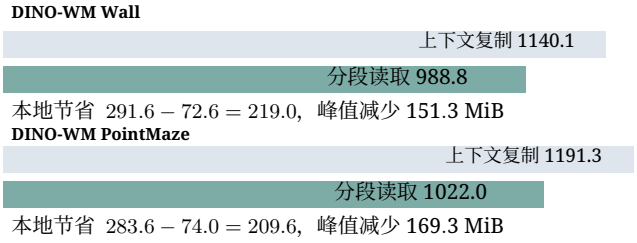

图 4. **从历史副本到全模型峰值**。注释标明了新增工作区减去移除的存储空间（MiB）。

表 6。 **固定精度下的设备限制**。

工作负载	限制	调优后	PTQ4WAM
进程峰值内存（MB）			
Fast-WAM Long	9,000	8,420	7,861
LeWM Push-T	850	752	724
DINO-WM Push-T	2,400	2,137	1,894
RoboCasa Windows	3,100	**3,388**	2,849
重放时间（秒）			
Fast-WAM Long	7.60	7.458	6.495
LeWM Push-T	4.00	3.748	3.245
DINO-WM Push-T	24.00	21.095	18.967
RoboCasa Windows	61.00	58.405	51.779

当执行变更影响模型是否符合条件时。我们针对每个冻结映射，依据设备限制测试了两条调优路径（表6）。即使在记录中最慢的100次调用重放中，两者也都满足时间限制。在三个工作负载中，两者也都能装入内存，因此，更快的实现为一个原本就符合条件的模型节省了时间。在RoboCasa Windows上，调优路径超出了3,100 MB的进程限制。直接读取和融合重缩放使同一映射符合限制。两条路径对每个记录窗口的结果都一致。

首选实现也可能随设备预算而改变。在短历史记录下，DINO-WM Wall 采用复制历史记录并分别运行各项操作的方式，执行速度最快。直接读取历史记录并融合重新缩放操作内存占用更少，但在该输入上的耗时更长。复制路径耗时 4.95 s，峰值为 972.4 MiB，符合 5.10 s 的截止时间和 985 MiB 的分配上限。在 840 MiB 的上限和 5.50 s 的截止时间下，只有组合路线符合要求，其内存占用为 821.6 MiB，耗时 5.33 s。因此，从复制路径改为组合路线可节省 150.8 MiB，但会使重放耗时增加 0.38 s。这些上限针对已分配张量。表中所示的进程峰值6 还包括运行时在这些张量之外持有的内存。

工作负载的变化也会影响选定的权重。我们将 Fast-WAM Goal 地图迁移到 Long 的 Long 字节预算，并将 LeWM Wall 地图迁移到 Push-T。在 Long 字节预算下，Goal 选择的地图为 614 个组件中的 369 个分配了四位，且未使用任何 Long 开发分数。在该预算下重新选择后，成功率从 92.13% 提高到 93.56%。Wall 地图使用的四位组件数量与 Push-T 目标相同，但针对目标重新选择也提升了该任务的成功率。在相同存储预算下，目标反馈使 Push-T 的成功率提高了 2.32 个百分点，使 Long 的成功率提高了 1.43 个百分点。两种情况下，直接迁移都无需目标试验，而重新选择则需要 12 次。在新设备上进行搜索，需要针对可用路径重新测量计时和内存。

我们还在固定任务和检查点的同时，测试了输入分布的偏移。在 Long 上，我们更改了起始状态和目标，并延长了时域。TwoRoom 的通道变窄了，而 Windows 录制数据则出现了亮度变化和部分遮挡。这三种偏移都降低了冻结映射的分数，不过在每个偏移任务上，其观测分数仍高于 FP32。使用 12 次目标试验重新进行选择后，分数比冻结映射提高了 4.60 至 5.90 个百分点。我们使用相同的偏移源来评估两种映射。这些增益衡量了来自偏移任务的反馈所带来的价值。

5. 结论

压缩世界动作模型时，必须在设备限制下保留决策。相似的特征误差可能导致不同的动作，而较小的权重仍可能超出 GPU 内存或延迟限制。PTQ4WAM 会逐层选择精度，并同时选择执行路径。校准引导的提案和全模型性能分析会在昂贵的任务试验之前筛选候选对。原始规划器或控制器会根据任务结果为可行配对评分。融合分组重缩放和直接历史读取在保持精度映射固定的同时，减少了临时存储空间。在来自四个模型系列的 15 种工作负载上，PTQ4WAM 保留了相应 FP32 任务得分中位数的 99.2%。固定输入的 GPU 重放显示，中位数 $2.36\times$ 加速比，以及峰值分配内存中位数 71.5% 的缩减。这些结果表明，评估压缩效果时，应考量任务决策以及执行模型所需的资源。

大语言模型使用声明

ChatGPT（OpenAI）协助润色英文、调整论文和补充材料中的 LaTeX 格式，并为实验提供了有限的辅助工作。在候选项提案实验中，Qwen3-8B 使用校准数据提出精度分配方案。确定性生成器会在任务评估前检查每项提案。作者独立提出了相关思路和方法，审阅了 AI 辅助生成的材料，并对论文内容负责。

参考资料

[1] Arash Akbari、Arman Akbari、Jingwu Luo、Yuhao Lei、Yi Gao、Weiwei Chen、Xuan Zhang、Zhenman Fang、Geng Yuan 和 Yanzhi Wang。《Q-WAM：通过动作子空间保护实现世界动作模型的 4 位量化》。arXiv:2609.33269，2026年。[2] Saleh Ashkboos、Amirkeivan Mohtashami、Maximilian L. Croci、Bo Li、Pashmina Cameron、Martin Jaggi、Dan Alistarh、Torsten Hoefler 和 James Hensman。《QuaRot：旋转大语言模型中无离群值的 4 位推理》。载于《神经信息处理系统进展》，2024年。[3] Mido Assran、Adrien Bardes、David Fan 等。《V-JEPA 2：自监督视频模型实现理解、预测与规划》。arXiv:2506.09985，2025年。[4] Tianqi Chen、Thierry Moreau、Ziheng Jiang、Lianmin Zheng、Eddie Yan、Haichen Shen、Meghan Cowan、Leyuan Wang、Yuwei Hu、Luis Ceze、Carlos Guestrin 和 Arvind Krishnamurthy。《TVM：面向深度学习的自动化端到端优化编译器》。载于 USENIX 操作系统设计与实现研讨会，第578–594页，2018。[5] PyTorch贡献者。《torch.compiler》。PyTorch编译器文档。访问日期：2026年9月25日，2026年。[6] Tri Dao、Daniel Y. Fu、Stefano Ermon、Atri Rudra 和 Christopher Ré。《FlashAttention：具备 IO感知能力的快速、内存高效精确注意力》。载于《神经信息处理系统进展》，2022年。[7] Zhen Dong、Zhewei Yao、Daiyaan Arfeen、Amir Gholami、Michael W. Mahoney 和 Kurt Keutzer。《HAWQ-V2：神经网络的海森感知迹加权量化》。载于《神经信息处理系统进展》，2020年。[8] Elias Frantar、Saleh Ashkboos、Torsten Hoefler 和 Dan Alistarh。《GPTQ：面向生成式预训练转换器的精确训练后量化》。载于国际学习表征会议，2023年。[9] Elias Frantar、Roberto L. Castro、Jiale Chen、Torsten Hoefler 和 Dan Alistarh。《MARLIN：大语言模型上的混合精度自回归并行推理》。载于 ACM SIGPLAN 并行编程原理与实践研讨会，2025年。[10] Zhongqian Fu、Tianyi Zhao、Kai Han、Hang Zhou、Xinghao Chen 和 Yunhe Wang。《世界模型量化的实证研究》。arXiv:2602.02110，2026年。[11] Jacob R. Gardner、Matt J. Kusner、Zhixiang Xu、Kilian Q. Weinberger 和 John P. Cunningham。《带不等式约束的贝叶斯优化》。载于国际机器学习会议，第937–945页，2014。

[12] Han Guo、William Brandon、Radostin Cholakov、Jonathan Ragan-Kelley、Eric P. Xing 与 Yoon Kim。面向查找表量化大语言模型的快速矩阵乘法。载于《计算语言学协会论文集：EMNLP 2024》，第12419–12433页，2024年。[13] Woosuk Kwon、Zhuohan Li、Siyuan Zhuang、Ying Sheng、Lianmin Zheng、Cody Hao Yu、Joseph E. Gonzalez、Hao Zhang 与 Ion Stoica。基于分页注意力的大语言模型服务高效内存管理。载于 ACM 操作系统原理研讨会，2023年。[14] Muyang Li、Yujun Lin、Zhekai Zhang、Tianle Cai、Xiuyu Li、Junxian Guo、Enze Xie、Chenlin Meng、Jun-Yan Zhu 与 Song Han。SVDQuant：通过低秩分量吸收离群值以实现4位扩散模型。载于国际学习表征会议，2025年。[15] Tianqi Li、Wenyu Fang、Xin He、Xue Geng、Xu Cheng 与 Yun Liu。当 W4A4 破坏伪装目标检测时：令牌组双约束激活量化。载于欧洲计算机视觉会议，第77–95页，2026年。[16] Yuhang Li、Ruihao Gong、Xu Tan、Yang Yang、Peng Hu、Qi Zhang、Fengwei Yu、Wei Wang 与 Shi Gu。BRECQ：通过区块重构推动训练后量化的极限。载于国际学习表征会议，2021年。[17] Ji Lin、Jiaming Tang、Haotian Tang、Shang Yang、Wei-Ming Chen、Wei-Chen Wang、Guangxuan Xiao、Xingyu Dang、Chuang Gan 与 Song Han。AWQ：面向端侧大语言模型压缩与加速的激活感知权重量化。载于《机器学习与系统会议论文集》，2024年。[18] Zechun Liu、Changsheng Zhao、Igor Fedorov、Bilge Soran、Dhruv Choudhary、Raghuraman Krishnamoorthi、Vikas Chandra、Yuandong Tian 与 Tijmen Blankevoort。SpinQuant：采用学习得到的旋转的大语言模型量化。载于国际学习表征会议，2025年。[19] Lucas Maes、Quentin Le Lidec、Damien Scieur、Yann LeCun 与 Randall Balestriero。LeWorldModel：基于像素的稳定端到端联合嵌入预测架构。载于《神经信息处理系统进展》，待发表，2026年。[20] Markus Nagel、Rana Ali Amjad、Mart van Baalen、Christos Louizos 与 Tijmen Blankevoort。向上还是向下？面向训练后量化的自适应舍入。载于国际机器学习会议，第7197–7206页，2020。[21] Lin Niu、Jiawei Liu、Zhihang Yuan、Dawei Yang、Xinggang Wang 与 Wenyu Liu。利用任务损失引导的 L_p 指标改进目标检测上的训练后量化。arXiv:2304.09785，2023年。[22] 英伟达。CUDA 中的高效通用矩阵乘法。CUTLASS 文档，访问日期：2026年9月25日，2026年。[23] Wenqi Shao、Mengzhao Chen、Zhaoyang Zhang、Peng Xu、Lirui Zhao、Zhiqian Li、Kaipeng Zhang、Peng Gao、Yu Qiao 与 Ping Luo。OmniQuant：面向大语言模型的全向校准量化。载于国际学习表征会议，2024年。[24] Sheng Shen、Zhen Dong、Jiayu Ye、Linjian Ma、Zhewei Yao、Amir Gholami、Michael W. Mahoney 与 Kurt Keutzer。Q-"

BERT：基于海森矩阵的 BERT 超低精度量化。载于AAAI人工智能大会，2020。[25] Fei Wang、Chao Xue、Taoran Liu、Li Shen、Ye Liu 与 ChangXing Ding。超越激活对齐：任务感知的大语言模型量化中的对齐-多样性权衡。arXiv:2607.00908，2026年。[26] Kuan Wang、Zhijian Liu、Yujun Lin、Ji Lin 与 Song Han。HAQ：采用混合精度的硬件感知自动量化。载于 IEEE/CVF计算机视觉与模式识别会议，2019年。[27] Yunhan Wang、Haodong Wang、Zhiming Liu、Zicong Hong、Qianli Liu、Xiaoyi Pang、Yangjia Hu、Quanxin Shou、Yikun Miao 与 Song Guo。SteerQuant：在世界动作模型中通过动作引导缩放引导量化误差。arXiv:2609.39056，2026年。[28] Xiuying Wei、Ruihao Gong、Yuhang Li、Xianglong Liu 与 Fengwei Yu。QDrop：面向极低比特训练后量化的随机丢弃量化。载于国际学习表征会议，2022。[29] Guangxuan Xiao、Ji Lin、Mickael Seznec、Hao Wu、Julien Demouth 与 Song Han。SmoothQuant：面向大语言模型的准确高效训练后量化。载于国际机器学习会议，第38087–38099页，2023年。[30] Hanqi Xiao、Yi-Lin Sung、Elias Stengel-Eskin 与 Mohit Bansal。任务电路量化：利用知识定位与可解释性进行压缩。载于语言建模会议，2025年。[31] Tianyuan Yuan、Zibin Dong、Yicheng Liu 与 Hang Zhao。Fast-WAM：世界动作模型是否需要测试时未来想象？载于《神经信息处理系统进展》，待发表，2026年。[32] Zhihang Yuan、Chenhao Xue、Yiqi Chen、Qiang Wu 与 Guangyu Sun。PTQ4ViT：采用双均匀量化的视觉变换器训练后量化。载于欧洲计算机视觉会议，第191–207页，2022。[33] Gaoyue Zhou、Hengkai Pan、Yann LeCun 与 Lerrel Pinto。DINO-WM：基于预训练视觉特征的世界模型实现零样本规划。载于国际机器学习会议，PMLR 267，第79115–79135页，2025年。[34] Jiacheng Zhou、Jinfan Lv、Ruixuan Li、Longtai Zhang、Yan Wang、Wenqiang Zhang 与 Lizhe Qi。QuantWAMs：为世界动作模型在合适粒度上进行校准。arXiv:2607.28405，2026年。

面向动作的量化，面向部署的执行：世界动作模型的精度与执行联合选择

补充材料

补充材料概览。附录A 介绍评估协议，包括源数据划分和工作负载设置。附录B 报告资源受限联合选择和匹配的实现控制。附录C–E详述精度分配、稳健性检查、候选池限制和准备成本。附录F–G 记录部署结果、存储组成和执行检查。附录H–I 说明测量研究的范围，并呈现补充对照比较。附录J 给出形式化分析和证明。附录K–M 提供重复测量、部署注册表和源记录，以及成对的质量与执行结果。附录N 和O最后介绍研究范围、可复现性细节，并讨论研究发现。

评估协议将模型选择与模型运行区分开来。第B 节在新源上执行一次映射（`-J`），重复整个搜索过程（`-JR`），并在图形处理器（GPU）上重放固定调用（`-X`）。第C–E 节记录选择和提案控制，包括分阶段反馈（`-CB`）。部署与执行记录给出源计数、计时边界和模拟器批次（`-Z`，`-ZR`）。证明阐述有限候选池和峰值内存结果背后的假设。最后几节将每对测量值与产生该结果的运行关联起来。

我们报告样本标准差（SD）、置信区间（CI）以及以百分点（pp）表示的差异。我们使用 p95 表示第 95 百分位数。对于每个源的记录，CAL、DEV 和 FINAL 分别标识用于校准、开发和最终评估轮次的数据集。

FP32 表示 32 位浮点数，BF16 表示脑浮点 16。W4 和 W8 分别指四比特和八比特权重，A8 和 A16 分别指八比特和十六比特激活值。我们以十进制兆字节（MB）或二进制兆字节（MiB）报告内存，并在每张表中标明所用单位。

A. 评估协议

A.1. 来源与估计量

表 S1列出了每个受控工作负载中的源。96 个选择区块中的每个区块都包含 64 个校准源、128 个开发源和 256 个最终源，因此每项任务共有 43,008 个不同的源。原生精度比较包含 48 个区块，每项任务有 24,576 个源。在这两种设计中，各划分和区块之间的来源标识符互不重叠。规划时，我们将同一初始状态的重复执行归在一起；对于 Franka Ranking，将同一回合中的比较归在一起；对于 RoboCasa Windows，则将同一轨迹中的窗口归在一起。

B001–B096 这些标签用于标识受控选择区块。任务选择器（S）和提案子集（N）的独立配对研究各有 12 个区块，并且各自有一份清单。64 次运行的执行研究以及由四项比较组成的系列也分别有各自的记录。校准输入（A）、等反馈（F）和排序执行（R）研究也各自保留独立的清单。附录 L.3中的源清单给出了这些协议各自独立的父级计数。E 在每个工作负载中保留 12 个实测执行对，这些执行对取自其自身记录的比较。附录 M给出了配对估计量。附录 J.4 说明了我们如何对用于选择界的源进行抽样。

表 S1. 受控源划分与结果单位。

系列	工作负载	每个区块的结果数
Fast-WAM	Long	768 个规划结果
LeWM	Wall-L、Push-L、TwoRoom、Cube	768 个规划结果 每个工作负载
DINO-WM	Push-D	768 个规划结果
V-JEPA 2-AC	排名	768 次比较
	窗口	1,024 个窗口

对于选择器 s、规划源 j 和执行 r，令$y_{s,\ j,\ r} \in \{0,1\}$。如果有 N 个独立来源，百分比得分将为每个来源赋予相同权重，

$$\bar{y}_{s,j} = \frac{1}{3}\sum_{r=1}^{3} y_{s,j,r}, \qquad Q_s = \frac{100}{N}\sum_{j=1}^{N} \bar{y}_{s,j}. \tag{S1}$$

重采样会保留规划源的全部执行次数、某个回合的比较结果，或从记录的源轨迹中提取的窗口。记录清单示例包括 Long/B001/CAL/FWL-C-000001、Long/B001/ DEV/FWL-D-000001 和 Long/B001/FINAL/FWL- E-000001。它们的来源组为回合，与之关联的轨迹窗口也共享其来源组。在部署队列中，每个规划源贡献一个二元结果。在选择器比较中，每个源贡献其三次规划执行的均值。

RoboCasa 上的窗口级质量。主论文表 1 中的 Windows 部署评估使用了 14 条记录的源轨迹，每条轨迹取 4 个短片段，共得到 56 个评估窗口。令 $r_{jk} \in \{0,1\}$ 表示轨迹 j 中窗口 k 的报告结果。我们对

所有窗口的二值评分取平均，得到

$$Q_{\mathrm{win}} = \frac{100}{56}\sum_{j=1}^{14}\sum_{k=1}^{4} r_{jk} = \frac{100}{14}\sum_{j=1}^{14}\left(\frac{1}{4}\sum_{k=1}^{4} r_{jk}\right).$$

每条轨迹都贡献相同的四个窗口。因此，汇总观测值所得的估计与对轨迹内比率取平均值相同。来自同一条轨迹的四个窗口始终归入同一个重采样单元。

A.2. 工作负载设置

表 S2列出了执行这些工作负载时使用的设置。Fast-WAM 对照组任务来自机器人操作任务终身学习基准（LIBERO）。

B. 资源可行的联合选择与实现控制

B.1. 匹配地图/路径选择消融

本实验改变搜索检查候选项执行路径的时机。质量优先选择（任务→路径）会对候选映射评分，按任务质量进行选择，然后确定路径。联合选择则将开发任务预算用于满足资源限制的候选项。在协议-J中，Fast-WAM Long、LeWM Push-T（Push-L）、DINO-WM Push-T（Push-D）和 RoboCasa Windows 使用四比特或八比特权重、八比特激活值以及大小为 32 的分组。受保护的 Windows 操作使用 16 比特激活值。每项任务使用 64 个校准源和 128 个开发源进行一次搜索。我们冻结所得的配对方案，并在 12 个互不重叠的区块上进行评估，每个区块包含 256 个源。每个源进行三次规划执行，或每条轨迹产生四个 Windows 结果，因此区块分母分别为 768 和 1,024。我们按区块配对结果，并将每对结果作为一个观测值。

使用随机种子 20261115，每个 48 映射池都包含 16 个映射，分别对应相对于已注册 `-R:{task}:C00` 基准映射的交换半径 1、2 和 4。在每个半径下，参数量相同的 W4/W8 交换按以下字符串的 256 位安全散列算法（SHA-256）摘要排序：`J:task:radius:attempt:i:j`。贪心选择互不重叠的交换，拒绝重复映射；尝试从零开始，最多进行 100,000 次。每次交换都保持参数数量和存储预算不变。包含 12 个映射的提案池，每个半径有四个映射，并在开发评分前冻结。最终结果不会参与提案或映射选择。PTQ4WAM、确定性提案和约束贝叶斯优化（贝叶斯优化）均获得相同的校准信息和 12 次任务评估。贝叶斯优化可以在同一个 48 映射池中自适应地进行开发评估。Pareto-48 会评估全部 48 个映射。映射标识符仅在 `-J` 范围内有效。每个任务都有各自的回退映射，并在相应候选池中注册为 `-J:{task}:C00`。表 S3比较所得映射的留出得分。表 S4记录了对部署限制的遵守情况，表 S5则给出了每个配对区块的底层计数。

回退制品在开发评分前，根据校准和资源规则固定下来。其构建和性能分析计入准备阶段，且不额外使用任务试验。如果任何受监控调用超出上限，我们就标记该区块使用了回退。只有受影响的源才会从相同的初始状态出发，使用回退制品重新运行。最终结果和失败尝试时间都会计入，但不会增加质量分母。相同规则适用于每种方法。全部十二个评估区块均使用该次搜索中选定的映射和路径。表 S6 使用配对区块比较 PTQ4WAM 与受约束贝叶斯优化。

Task→route 和 Heuristic-12 的汇总成功次数分别为：Long 为 8634/9216 和 8566/9216，Push-L 为 8224/9216 和 8117/9216，Push-D 为 8384/9216 和 8341/9216，Windows 为 4497/12288 和 4405/12288。这些汇总端点以点估计的形式报告。

设 c_i 为某种方法在区块 i 的 m个观测中取得的成功结果数。其汇总得分为 $Q = 100\sum_{i=1}^{12} c_i/(12m)$。对于 PTQ4WAM 和贝叶斯优化，我们分别用 c_i^{W}和 c_i^{BO}表示区块计数。两者配对差值的百分点为 $d_i = 100(c_i^{\mathrm{W}} - c_i^{\mathrm{BO}})/m$。我们使用¯个差值的均值 d 和样本标准差 s_d，计算样本方差时分母为 11。同时区间为 $\bar{d} \pm 2.9808724301\, s_d/\sqrt{12}$。在记录的有限候选池中，所有四个下界均大于零。我们为 S 协议和 N 协议分别抽取了样本。

贝叶斯优化比较固定了反馈预算。任务→路径监控器会统计发生资源违规后调用回退的运行次数。在这些有限候选池中，PTQ4WAM 以减少 75% 的任务评估次数达到与 Pareto-48 相同的表现。同一映射的执行效果在 -X 中单独测量。

B.2. 独立完整搜索比较（`-JR`）

在这里，每个观测都是一次新的搜索。对于四种工作负载中的每一种，我们都会构建候选池、校准模型、测试并选择映射，然后运行十二次最终评估。在每次重复中，所有策略共用 64 个校准父级源、128 个开发父级源和 256 个留出父级源。各次重复使用不相交的父级。一个规划父级贡献三次执行，而一个 Windows 父级贡献四个结果，因此每次搜索会产生 768 或 1,024 个最终观测。对于给定工作负载，覆盖全部十二次搜索的总计为 768 个校准父级、1,536 个开发父级和 3,072 个最终父级。

搜索从三个独立构建、序列化字节预算相同的映射开始。在每个具有相同数量符合条件参数的单元类别中，每个起始项的 W4 集合所含计数由附录中序列化注册表指定L。符合条件的单元标识符（ID）按其美国信息交换标准代码（ASCII）的 SHA-256 摘要排序

表 S2. 各工作负载和协议的执行设置。

工作负载	记录的设置
Fast-WAM / LIBERO	48 个质量区块，每个区块 512 个源，每个源对应一个结果，64 次配对固定工作量计时运行，以及 20 次预热运行。
LeWM	48 个质量区块，每个区块 512 个源，每个源对应一个结果，64 次配对固定工作量计时运行，以及 20 次预热运行。
DINO-WM / Wall、PointMaze、Push-T	48 个质量区块，每个区块 512 个源，每个源对应一个结果，64 次配对固定工作量计时运行，以及 20 次预热运行。
DINO-WM / Meta-World Reach	48 个质量区块，每个区块 512 个源，每个源一个结果，64 次配对固定工作量计时运行以及 20 次预热运行。
V-JEPA 2-AC / 排序	48 个质量区块，每个区块 512 个回合，且每个回合三次比较，64 次配对固定工作量计时运行以及 20 次预热运行。
V-JEPA 2-AC / 窗口	48 个质量区块，每个区块包含 512 条轨迹，且每条轨迹有四个窗口；64 次配对固定工作量计时运行和 20 次预热运行。
-J，四项任务	一次 64 源校准和 128 源开发集选择，随后进行 12 个冻结映射最终评估区块，每个区块包含 256 个源；每个规划源有三个结果，或每条 Windows 轨迹有四个窗口。
-X，四项任务	20 次预热、12 次配对重放、每次重放 100 次模型调用、批次大小为 1、上下文长度为 64/8/16/64、不含环境步数（附录 B.4）。
-Z，Long	500 个回合，每回合 180 个环境步数，动作块大小为 12，共 7,500 次模型调用，完整模拟器流水线时序（附录 I）。

表 S3. 冻结搜索实验中的留出集质量（%）。

Task	任务→路径	Heuristic-12	BO-12	Pareto-48	PTQ4WAM-12
Long	93.6849	92.9470	94.0213	95.9744	95.9744
Push-L	89.2361	88.0751	89.5074	92.0139	92.0139
Push-D	90.9722	90.5056	91.4280	93.5330	93.5330
Windows	36.5967	35.8480	37.2070	40.4053	40.4053
任务评估	12	12	12	48	12

全部策略共享相同的来源队列和资源上限。Pareto-48 和 PTQ4WAM-12 为每项任务选择了相同的映射。

表 S4. 进程内存和 100 次调用的 p95 上限。

Task	进程上限（MB）	回放 p95 上限（秒）	任务→路径可行	回退区块
Long	9000	7.60	10/12	2/12
Push-L	850	4.00	10/12	2/12
Push-D	2400	24.00	10/12	2/12
Windows	3100	61.00	11/12	1/12

回退前先统计可行性。字节　　O-12、Pareto-48 和 PTQ4WAM-12 均保持可行　　每项任务全部 12 个区块。

字符串 -JR |任务 |随机种子 |起始项 |类别 |单元，然后按单元 ID 排序。每个起始项在交换半径 1、2、4 和 8 下各提供四个映射。对于半径和尝试次数 0–99999，符合条件且参数相同的 W4/W8 交换对按 -JR |任务 |随机种子 |起始项 |半径 |尝试次数 |i |j 的 SHA-256 摘要排序，然后按 $(i,\ j)$ 排序。随后贪心地选取互不相交的配对。字段使用原始任务/单元标识符、无前导零的十进制整数，且不含空格或末尾换行符。在所有起始项和半径中，48 个已接受映射 C00–C47 具有互不相同的有效位集、精确的字节预算（相对误差为零），并且至少有 12 个资源可行对。候选池耗尽或可行配对不足时，搜索即告结束。我们绝不使用留出数据来抽取替代候选池。

所有方案均使用 RTX 4090、共同编译的后端、组大小为 $G = 32$ 的 W4/W8-A8（Windows 也采用受保护的 A16 运算）、输入形状、批大小为一、表 S7 中的上下文和上限，以及相同的调优预算。48 个映射均在校准探针上通过两条固定路径进行评估。一条将上下文复制到缓冲区，另一条读取原始片段。对于每个映射/路径，进行 20 次预热和 100 次同步计时的完整模型调用，以得出经验秩 95 的 p95；进程内存峰值则在相同输入约定下读取：$48 \times 2 \times (20+100) = 11520$ 每次搜索进行的性能分析调用。路线切换、数值检查、调优、失败和打包均计入完整准备。性能分析会尽可能选择速度最快且满足上限约束的路线；否则保留速度最快的数值有效诊断路线。12 次任务试验留给通过可行性检查的配对。

PTQ4WAM 和受约束贝叶斯优化各对 12 个可行

表 S5. 冻结搜索的配对最终结果计数。

	Long		Push-L		Push-D		Windows	
区块	BO	PTQ4WAM	BO	PTQ4WAM	BO	PTQ4WAM	BO	PTQ4WAM
1	713	729	698	715	700	717	407	438
2	730	740	681	710	703	719	415	434
3	722	736	723	727	701	697	356	401
4	718	737	684	705	697	712	389	407
5	727	738	683	701	699	716	375	413
6	731	742	694	713	708	731	403	449
7	721	735	667	695	715	732	366	390
8	726	748	676	698	690	715	362	395
9	713	725	682	698	689	706	387	411
10	721	726	673	697	702	714	384	409
11	716	744	699	708	705	721	343	397
12	727	745	689	713	717	740	385	421

每个规划区块包含 768 个结果，每个 Windows 区块包含 1,024 个结果。

表 S6. 搜索一次后 PTQ4WAM 与受约束贝叶斯优化的比较。

Task	Mean	标准差 同时 95%置信区间	Holm p
Long	1.9531 0.7969	[1.2674, 2.6388]	7.38745×10^{-6}
Push-L	2.5065 0.9473	[1.6913, 3.3217]	5.25891×10^{-6}
Push-D	2.1050 0.9614	[1.2778, 2.9323]	1.08005×10^{-5}
Windows	3.1982 1.1167	[2.2374, 4.1591]	3.19904×10^{-6}

差值和样本标准差以百分点（pp）表示。区间采用四任务 Bonferroni 校正，配对-t 检验采用 Holm校正。

配对进行评分，共进行 $12 \times 128 = 1{,}536$ 次候选项–父项访问。两种策略都会接收完整的校准测量数据、全部48个映射、每个映射对应的两个路径配置文件，以及用于在开发数据上评分的相同目标函数。PTQ4WAM 按校准确定的顺序检验提案。贝叶斯优化使用运行随机种子初始化四个可行配对，然后通过受约束期望改进提出八个配对。其 Matérn-5/2高斯过程使用标准化位模式和资源特征，以及已完成的开发集得分。若预测结果并列，则候选项标识符较早的映射排在前面。

质量优先选择采用与 PTQ4WAM 相同的校准排序。它先填满 12 个名额，再应用可行性筛选，为这些候选项评分，然后检查胜出的配对。校准已知违规会触发预先声明的回退。若在最终父项上新发现违规，则记为一次失败尝试，并在该父项上重新运行。全部 48 次质量优先搜索在留出数据源上测试之前，均选出了校准可行的胜出项。穷举基准会为全部 48 个预先确定的配对评分（$48 \times 128 = 6{,}144$ 次访问），并选出其中最佳的可行配对。若只对 F 个可行配对进行穷举评分，则需要 $128F$ 次访问。获得所有可行配对的分数后，在相同平局规则下，两种检查顺序都会选出同一个最佳可行项。名额为 12 时，检查会改变哪些配对获得评分。表 S8比较最终得分，表 S9 报告配对差值。表 S10 和S11 给出了全部计数四种工作负载。表 S12 展示了哪些搜索在留出源上不得不使用回退。

在四个系列中，PTQ4WAM 相较于贝叶斯优化的均值增益为 2.0182–2.7995 个百分点。PTQ4WAM 相较于指定的质量优先消融方法，增益为 1.7578–2.4251 个百分点。八项双侧配对比较采用单一的 Holm 八项系列，并使用 $t_{1-0.05/16,11}$ 同时置信界。符号检验提供了第二种配对分析。对于 PTQ4WAM 相较于 Pareto-48审计的比较，采用 $t_{1-0.05/4,11}$ 得出的四个同时单侧下界均超过 -0.50 个百分点的非劣效性界值。各方法的排名逆转仍保留在运行数组中。Pareto-48 是一项固定的 48 对审计。

运行与数据源绑定。 运行-01 `-JRLong` 智能体记录使用 Qwen3-8B（温度 0.6，top-p 0.9）、随机种子 104729、校准输入 `-JR-Long-01-cal.json`，以及附录 L.3中的提示词约定。其返回状态正常，必需W8列表和首选W4列表均为空，$n_4^{\min} = n_4^{\max} = 369$，$d_{\mathrm{H}} \geq 2$，以及候选标识符C01、C04、C09、C15、C17、C22、C24、C30、C33、C36、C42、C46。修复过程中，C48这一无效项被C46替换一次，且未使用回退。代码`-JR:Long:01:P48` 标识了用于进行此次选择的候选池。

关联的 PTQ4WAM 记录是运行 -JR:Long:01:W，使用检查点 -Fast-WAM-r1。符合条件的单元列于 -Fast-WAM-units-r1。此运行使用代码版本 -code-r1 和已编译内核版本 -compiled-r1。我们将配置记录在 -JR-Long-config 下。对于源区块 -JR:Long:01，选定映射为 C17，其路径为 -r02。此映射的位集为 - JR- Long- 01- C17- bits。模型和日志分别使用不同的键。我们用 - JR- Long- 01- C17- artifact 标识保存的模型，并用 - JR-Long-01-W-log 标识日志。预算和序列化制品均为 3,888,653,312 字节，而连接后的

表 S7。独立完整搜索的资源限制。

Task	预算（字节）	进程 MB	调用 p95（秒）	上下文	T 可行	B/W/P 可行
Long	3888653312	9000	0.085	64	8/12	12/12
Push-L	9212928	850	0.045	8	9/12	12/12
Push-D	58282752	2400	0.280	16	10/12	12/12
Windows	430250240	3100	0.680	64	9/12	12/12

表 S8. 最终质量与有限总体非劣效性。

Task	T	B	W	P	W–P（百分点）	非劣效性下限（百分点）
Long	93.3811	93.1315	95.1714	95.2365	-0.0651	-0.2120
Push-L	89.7569	88.7695	91.5582	91.6992	-0.1411	-0.2991
Push-D	91.8837	91.6233	93.6415	93.7500	-0.1085	-0.2453
Windows	37.5244	37.1501	39.9495	40.0798	-0.1302	-0.2397

T 表示质量优先选择，B 表示受约束贝叶斯优化，W 表示 PTQ4WAM，P 表示穷尽式审计。NI 表示非劣效性。同时单侧 95% 下界采用 -0.50 百分点的非劣效界值。

表S9. 独立搜索之间的配对差值。

Task	对比	均值 ± 标准差（百分点）	同时置信区间（百分点）	Holm t p	Holm符号检验 p
Long	W–BO	2.0399 ± 0.8445	[1.2184, 2.8615]	1.27×10^{-5}	0.019043
Long	W–T	1.7904 ± 0.4682	[1.3348, 2.2459]	2.1×10^{-7}	0.003906
Push-L	W–BO	2.7886 ± 0.5914	[2.2133, 3.3640]	2.78×10^{-8}	0.003906
Push-L	W–T	1.8012 ± 0.5023	[1.3126, 2.2899]	3.26×10^{-7}	0.003906
Push-D	W–BO	2.0182 ± 0.9463	[1.0976, 2.9389]	2.76×10^{-5}	0.019043
Push-D	W–T	1.7578 ± 0.3662	[1.4016, 2.1141]	2.68×10^{-8}	0.003906
Windows	W–BO	2.7995 ± 1.5171	[1.3235, 4.2754]	5.13×10^{-5}	0.038574
Windows	W–T	2.4251 ± 0.3455	[2.0890, 2.7612]	5.21×10^{-10}	0.003906

十二次完整搜索 次/任务。这些区间同时涵盖八项双侧比较。两项检验均报告经调整的 p 值 s.

表 S10。Long 和 LeWM Push-T 的完整搜索计数。

		Long				Push-L			
Run	Seed	T	B	W	P	T	B	W	P
01	104729	721	717	737	740	692	682	706	705
02	130363	708	698	722	724	683	671	692	693
03	155921	720	717	730	729	671	675	687	686
04	179021	733	735	750	753	697	685	708	707
05	203807	727	720	736	736	705	698	723	723
06	229393	714	717	732	733	689	672	699	702
07	256019	709	705	721	721	700	693	716	719
08	283051	716	731	729	728	689	681	698	699
09	310019	720	723	737	736	676	677	696	697
10	338021	715	704	726	727	693	681	709	712
11	367027	714	714	733	732	692	682	702	703
12	397027	709	702	718	718	685	684	702	705

每次运行中，四个组的每个规划工作负载都共享相同的768个最终结果。

_JRraw.Long.W.run01 单元为 737/768。符合条件的单元 .u000 将 .encoder.block0.q_proj.weight 绑定到 BF16 形状 $[2048, 2048]$, 打包偏移量 $[0, 4194304)$、图节点 .matmul000，且没有别名。此运行中最大的权重绝对值为 1.372903。我们使用广义高斯–牛顿曲率的对角估计，测得校准敏感度为 0.00849127。日志中估计器的名称为 GGN-diagonal-32。

清单 - JR- Long - 01 - CAL 列出了本次运行中用于校准的输入。

父级 ID 遵循 -JR:task:run:split:p0000 的格式，其中 task 为 {Long, Push-L, Push-D, Windows}，run 为 01– 12，CAL 父级索引为 0000–0063，DEV 为 0000–0127，FINAL 为 0000–0255。规划的子级索引为 0–2，Windows 的子级索引为 0–3。在运行 01 中，源 -JR:Long: 01:FINAL:p0000:r2 属于键为

表 S11. DINO-WM Push-T 和 Windows 的完整搜索计数。

		Push-D				Windows			
Run	Seed	T	B	W	P	T	B	W	P
01	104729	699	698	710	709	380	374	407	410
02	130363	709	702	725	726	370	397	393	395
03	155921	695	712	708	710	395	393	427	426
04	179021	707	711	724	723	389	375	414	414
05	203807	712	706	721	721	390	371	411	413
06	229393	701	702	715	715	396	380	423	425
07	256019	704	695	720	720	384	370	406	409
08	283051	708	701	719	722	376	361	397	397
09	310019	712	712	730	730	390	420	418	419
10	338021	709	703	720	721	380	378	402	405
11	367027	713	710	727	730	384	380	412	414
12	397027	699	692	711	713	377	366	399	398

Push-D 每次运行有 768个最终结果，Windows 每次运行有 1,024 个。

表 S12. 独立的质量优先搜索中的回退使用情况。

	Long		Push-L		Push-D		Windows	
Run	父节点	后继节点	父节点	后继节点	父节点	后继节点	父节点	后继节点
01	18	52	0	0	0	0	0	0
02	0	0	0	0	0	0	14	20
03	0	0	12	30	17	45	0	0
04	29	84	0	0	0	0	0	0
05	0	0	0	0	0	0	26	40
06	13	35	0	0	0	0	0	0
07	0	0	21	58	0	0	0	0
08	0	0	0	0	0	0	0	0
09	24	66	16	41	0	0	0	0
10	0	0	0	0	0	0	18	28
11	0	0	0	0	0	0	0	0
12	0	0	0	0	23	62	0	0

每对列出受影响的父节点 以及回退成功次数。质量优先总计包括这些成功次数。表S110 还给出了回退时的比率 ck 分数为零 .

- JR:Long:01:FINAL:p0000 的父级。若要查找此观测对应的会话，我们使用键 - session- Long- 01- FINAL- 0000。另一个键 - state- Long- 01- FINAL-0000 给出了源的初始状态。该源在运行块 01 中记录的结果为零。父级 ID 在不同协议数据划分或完整搜索之间不会重叠。我们使用连接来识别每个计数结果，并使用父级字段对来自同一搜索的结果进行分组。

B.3. 匹配的任务感知方法与内核实现

对于表 S13 中的任务感知方法改编，我们对 SteerQuant、QuantWAMs 和 Q-WAM 进行了改编 [1, 27,34]。Fast-WAM Long 和 Goal 实验使用 RTX 4090、四位或八位权重、八比特激活值和 32 个一组的分组设置。规划器、源数据划分以及每种方法的 12 次任务评估均保持一致。目标文件大小分别为 Long 的 4,110,417,920 B 和 Goal 的 2,923,963,392 B，相对容差为 10^{-3}。这项单独的比较有自己的 Long 存储目标。该文件包含动作感知组件和受保护分支。每个制品的序列化大小均包含缩放因子、填充和头部。

表 S13. 两项 Fast-WAM 任务上的任务感知训练后量化。

Task 方法	质量(%)	精确制品(B)
Long SteerQuant 适配版	92.83	4110528768
Long QuantWAMs 适配版	93.16	4111835136
Long Q-WAM 适配版	92.71	4111062016
Long PTQ4WAM	95.37	4110283008
目标 SteerQuant 适配版	95.61	2924217088
目标 QuantWAMs 适配版	95.98	2925278208
目标 Q-WAM-adapted	95.42	2923554304
目标 PTQ4WAM	97.43	2923963392

命名内核比较使用 RTX 4090、W4A16、组大小 128、相同的精度映射、形状、输入和调优上限，并采用完整的固定工作执行端点。

这些 W4A16/$G=128$ 路径保留各自的格式。该–x 编译适配器使用 A8/$G=32$。全部四条路径的任务质量相当（$\Delta Q=0$ 个百分点）。比较保留了每种实现所支持的数据类型、形状、融合设置、调优试验和回退行为。路径标识符指定历史消费者使用的实现。表 S14给出了原生 Marlin 内核及匹配实现的结果。

主论文第 4.4 节所追踪的步骤在预测器注意力处达到进程内存峰值。这些站点和启动计数适用于该步骤。表 S15 每次仅改变冻结映射上的一个输入分组参数。

B.4. 匹配重放与执行归因

主论文表 5 对所有工作负载均使用单次搜索实验的冻结胜出映射（–j）。固定调用重放（–x）使用四比特或八比特权重、八比特激活值以及大小为 32 的分组。受保护的 Windows 操作保留 16 比特激活值。每次重放执行 100 次模型调用，批大小为一，其中包括视觉或状态编码、预测或动作生成，以及读取历史记录的运算符。输入固定。Long、LeWM Push-T、DINO-WM Push-T 和 Windows 对应的上下文分别为 64、8、16 和 64。模拟器和停止规则位于此计时边界之外。编译后的低比特比较采用相同的格式和调优预算。两个 GPU 边界均经过同步。在进行 20 次不计时调用后，我们按照平衡顺序 AB、BA、BA、AB、AB、BA、AB、BA、BA、AB、BA、AB 收集 12 次配对重放（随机种子 20261116）。A 为编译适配器，B 为 PTQ4WAM。准备计时涵盖校准、获取反馈、模型打包、路由性能分析、编译和内核调优所需的时间。失败尝试也计入总成本。表S16 给出了重放端点，表 S17将冻结路径与三个执行基线进行比较。表 S18 列出了时间，表 S19 则使用这些配对重放比较两条路径。

BF16 的最终计数分别为 8772/9216、8418/9216、8484/9216 和 4409/12288。匹配的 PTQ4WAM 计数分别为 8845/9216、8480/9216、8620/9216 和 4965/12288。这些质量比较是汇总点估计。三个内存峰值分别测量，均未附带置信区间。

对于$d_i = T_{A,\,i} - T_{W,\,i}$，逐点区间为 $\bar{d} \pm 2.200985\,16\,01\, s_d/\sqrt{12}$，百分比节省量为$100(1 - T_W/T_A)$。四项计时检验构成各自的 Holm 检验族。在 $n = 12$处，最近秩经验 p95 即为观测到的最大重放时间。–x 数组与 E 及 64 次运行的执行表分开分析。表S20分别说明两项执行变更对时间和内存效应的影响。

在任务结果相同的情况下，与编译后的适配器相比，所选路线将平均重放时间缩短了 10.7942–13.4300%。 $q_{00}/q_{10}/q_{01}/q_{11}$ 控制项将执行干预单独隔离出来——同时保持选定映射固定。对于冻结的 Windows 映射，适配器进程峰值为 3388 MB，超过 3100 MB 的 -J 上限；而 PTQ4WAM 使用 2849 MB，并以 51.779 秒满足 61.00 秒的重放 p95 上限。对于此映射，路线的选择决定了设备能否同时满足两个上限。

B.5. 冷启动准备与完整覆盖台账

表S21分解了准备 Long 的一次性成本。表S22 比较了该成本与模型就绪后重放相同工作负载所需的时间。

对于路径 j, $T_{\text{total},\,j}(N) = T_{\text{prep},\,j} + NT_{\text{run},\,j}$。求解总时间首次低于该值时的重放次数，可得

$$N_{\text{strict}} = \left\lfloor \frac{7309.5 - 48.6}{10.326 - 6.352} \right\rfloor + 1 = 1828 \text{ replays.}$$

准备总耗时包括用于选择映射的任务试验。另一项模拟器实验在 25 个批次后达到盈亏平衡点。表S23 统计了全部 15 个部署工作负载中符合条件和已覆盖的算子。

B.6. 候选池大小与先验敏感性

独立的 Long :p 实验使用冻结的仅校准 C00，并采用随机种子为 20261115 的 NumPy PCG64。试验 j 使用半径 $(1、2、4、8)[j \bmod 4]$, 选择 r 个不同的 W4 索引，并按 W4 索引升序，选择参数数量相同且未使用的 W8 索引。若配对不可用、预算或约束无效，或位集重复，则拒绝该试验。接受的映射按试验顺序序列化，从 C00 开始。前 48、96 和 192 个映射构成嵌套候选池。搜索上限为 100,000 次试验，修复上限为 2，相对字节容差为 10^{-3}，未完成的候选池视为不可行。候选池清单在开发评分前冻结。重复计数为零，且 Agent-12 是 Broad-48 的子集。此多半径协议与附录 L 中的单交换候选池相互独立。每个候选池大小组使用 384 个开发结果。分别测量每个候选池的最终质量和准备时间。表S24 展示了候选池扩大时胜出映射的变化。表 S25 展示了名称和规则的选择如何改变待评估的映射。

固定预算下校准输入的影响。独立的 Long protocol :A 比较测量统计数据、打乱后的统计数据、移除的统计数据和确定性启发式方法；每种方法都在相同的字节预算和候选元数据下为 12 个候选项评分。Agent 条件保留单元名称。打乱操作仅在形状兼容组内置换校准统计数据，移除操作则省略其值。各方法在相同的最终父项上进行配对，涉及 12 个互不重叠的源区块，每个区块包含 256 个父项，且每个父项有三个规划结果（共 768 个结果）。配对

表 S14。组大小为 128 时匹配的 W4A16 内核。

路径	运行（秒）	第95百分位（秒）	已分配 MB	已保留 MB	进程 MB
CUTLASS 适配器	7.284	7.73	7684	7998	8734
Marlin 原生实现	7.103	7.51	7497	7812	8546
Marlin+分段消费者	6.936	7.37	7341	7655	8390
PTQ4WAM 路由	6.514	6.94	6996	7298	8022

表 S15. 固定权重下的单因素计时扫描。

上下文批次分组			适配器（μ秒）	PTQ4WAM（μ秒）
16	1	128	58.72	56.31
256	1	128	229.63	197.48
64	4	128	346.29	302.17
64	1	32	112.84	98.63

参考设置为上下文长度 64、批大小 1、组大小 128。更改实现不会改变任何一行的任务得分。

区块是统计单位（$n = 12$）。表 S26 列出全部结果。表 S27 报告经 Holm 校正的三项比较的双侧配对t 检验。每个区间对于其对应的比较均具有 95% 的覆盖率。

C. 精度分配与选择过程

对于工作负载 w，令 $\mathcal{U}_w = \{u_1, \ldots, u_{N_w}\}$ 表示符合条件的组件。组件 u_i 有 n_i 个权重，并分配到 $b_i \in \{4, 8\}$ 位。对于映射 $c = (b_1, \ldots\ldots, b_{N_w})$，令 $A_w(c)$ 表示保存的模型，令 $B_w > 0$ 表示其以字节为单位的目标大小，并令 $\mathcal{C}_w$ 表示合法候选映射。我们对符合条件的组件求和，以得到以字节为单位的完整文件大小，

$$S_w(c) = \sum_{i=1}^{N_w} \left\lceil \frac{n_i b_i}{8} \right\rceil + S_w^{\text{aux}}(c). \tag{S2}$$

在公式（S2）中，主论文公式（1）中的辅助项涵盖缩放因子、零点、受保护的运算、填充和头部。只有当映射的序列化大小满足以下条件时，才会将其纳入固定预算比较

$$|S_w(c) - B_w|/B_w \le \varepsilon_B, \qquad \varepsilon_B = 10^{-3}. \tag{S3}$$

对于固定路径，我们选择在相同开发源上得分最高的映射，

$$c_T = \arg\max_{c \in \mathcal{C}_w} \widehat{Q}_w^{\text{dev}}(c). \tag{S4}$$

在公式（S4）中，符号 $\widehat{Q}_w^{\text{dev}}(c)$ 省略了主论文公式（3）中所有候选项共有的路径。并列时，按照评估前确定的顺序进行选择。对于重构和海森矩阵评分，我们分别选择在同一候选池中使相应误差最小的映射，

$$c_R = \arg\min_{c \in \mathcal{C}_w} E_R(c), \qquad c_H = \arg\min_{c \in \mathcal{C}_w} E_H(c). \tag{S5}$$

算法 S1 映射与路径的联合选择**要求：** 预训练模型和工作负载。互不重叠的校准、开发和最终数据源。存储空间、进程内存、延迟和任务评估次数的限制。1：使用校准数据拟合量化统计量。使用这些数据按提出的顺序生成映射。2：**对** 每个有效映射**执行** 3：打包权重并检查完整文件大小。4：在校准输入上分析每条支持的路径。检查数值一致性、进程内存峰值和尾延迟。5：**结束循环**6：**如果** 没有路径满足限制**则**7：**返回**不可行。8：**结束条件**9：固定这些路径，并选取预算内的可行配对。在完整搜索中，每个映射使用其可用的最快可行路径。10：在开发数据上测量每个选定配对的任务得分。仅当两条路径在整个任务计算中产生确定性完全相同的输出时，才复用得分。11：保留得分最高的配对。并列时，依次按照延迟、进程内存和候选项顺序进行裁决。12：在留出数据源上测试之前，固定映射及其路径。准备阶段的成本包括性能分析、失败尝试、打包、调优以及为候选项评分所花费的时间。

附录 D.1 定义公式（S5）中的两种误差。主论文中的公式（4）允许选定映射携带其自身的可行路径。在联合搜索中，决策变量包含路径，并且在收集反馈之前应用主论文中的公式（2）。算法 S1 给出了路径检查以及用于选择最终配对的步骤。

生成可选子集。 令 I_w 包含模型结构、校准测量、存储预算和支持的格式。我们固定提示词和提案模型 ϕ。随后，随机抽样 ξ 确定提供给确定性映射生成器 Γ_w的约束 $\sim\Theta_w$：

$$\widetilde{\Theta}_w = \mathcal{A}_\phi(I_w, \xi), \qquad \mathcal{C}_w^A = \Gamma_w(\widetilde{\Theta}_w, I_w), \qquad |\mathcal{C}_w^A| = 12. \tag{S6}$$

四位组件数量的上限以及已接受映射之间的最小汉明距离 $d_{\min}$ 给出了以下约束，其中 $\mathbf{1}[\cdot]$ 在

表 S16. 匹配的重放时间、质量和内存（十进制 MB）。

Task	路径	质量（%）	时间（秒）	分配量	已保留	进程
Long	BF16-opt	95.1823	10.326000	11270	11560	12380
Long	Packed-same-map	95.9744	8.071000	7681	8002	8842
Long	编译适配器	95.9744	7.337417	7350	7690	8420
Long	PTQ4WAM	95.9744	6.352000	6846	7128	7861
Push-L	BF16-opt	91.3411	5.082000	271.4	341.2	864
Push-L	Packed-same-map	92.0139	3.958000	197.2	257.8	780
Push-L	编译适配器	92.0139	3.610500	179.6	232.4	752
Push-L	PTQ4WAM	92.0139	3.199083	161.3	216.7	724
Push-D	BF16-opt	92.0573	28.640000	3440	3730	4450
Push-D	Packed-same-map	93.5330	22.170000	1723	2035	2664
Push-D	编译适配器	93.5330	20.917833	1142	1450	2137
Push-D	PTQ4WAM	93.5330	18.659917	936	1234	1894
Windows	BF16-opt	35.8805	80.420000	3981	4263	4924
Windows	Packed-same-map	40.4053	61.910000	2810	3071	3756
Windows	编译适配器	40.4053	57.544583	2490	2732	3388
Windows	PTQ4WAM	40.4053	51.117500	2206	2416	2849

表 S17. 与三种执行基线的固定映射比较。

系列 / 任务	ΔQ 相比 BF16（百分点）	BF16 加速比	分配量 ↓(%)	进程 ↓(%)	打包加速比	适配器时间 ↓(%)
Fast-WAM / Long	+0.7921	1.6256×	39.2547	36.5024	1.2706×	13.4300
LeWM / Push-L	+0.6727	1.5886×	40.5674	16.2037	1.2372×	11.3950
DINO-WM / Push-D	+1.4757	1.5348×	72.7907	57.4382	1.1881×	10.7942
V-JEPA 2-AC / Windows	+4.5247	1.5732×	44.5868	42.1405	1.2111×	11.1689

加速比为基线时间除以 PTQ4WAM 重放时间。内存缩减相对于优化的 BF16 计算。适配器节省量采用其均值时间。Th e 报告的质量差异 d 描述观察到的分数。

表 S18. 配对的 100 次调用重放时间（秒） .

	Long		Push-L		Push-D		Windows	
重放	A	W	A	W	A	W	A	W
1	7.325	6.383	3.548	3.166	21.095	18.967	58.217	50.942
2	7.364	6.380	3.635	3.212	20.797	18.637	57.811	51.668
3	7.295	6.313	3.589	3.156	21.094	18.857	57.422	50.806
4	7.259	6.233	3.654	3.191	21.022	18.863	56.841	50.844
5	7.458	6.478	3.608	3.243	21.047	18.567	58.139	51.779
6	7.425	6.495	3.552	3.173	21.004	18.639	56.629	50.882
7	7.332	6.352	3.600	3.212	20.790	18.658	57.107	51.336
8	7.295	6.205	3.625	3.173	20.701	18.653	57.410	50.987
9	7.372	6.337	3.748	3.245	21.023	18.554	57.271	51.021
10	7.348	6.408	3.587	3.192	20.805	18.379	57.496	51.044
11	7.302	6.327	3.571	3.211	20.680	18.393	58.405	51.406
12	7.274	6.313	3.609	3.215	20.956	18.752	57.787	50.695

A 表示编译适配器，W 表示 PTQ4WAM。

其条件成立时取值为一，不成立时取值为零，

$$n_4^{\min} \le \sum_i \mathbf{1}[b_i = 4] \le n_4^{\max},$$
$$d_{\mathrm{H}}(c, c') = \sum_i \mathbf{1}[b_i \neq b'_i] \ge d_{\min} \quad (c \neq c'). \tag{S7}$$

公式（S7）中的距离采用主论文公式（5）的汉明指标。它统计位宽不同的分量。生成器会检查这些约束以及公式（S3）之后再接受映射。

逻辑上下文与活动分配。设$\mathbf{X}_1$，……，$\mathbf{X}_J$为上下文片段，ℓ_j 为它们沿上下文维度的长度，并令 $o_j = \sum_{r=1}^{j} \ell_r$ 满足 $o_0 = 0$。我们定位该分段，并读取以下偏移处的源元素，

$$\mathbf{X}_{\mathrm{virtual}}(p) = \mathbf{X}_j(p - o_{j-1}) = \mathbf{X}(p), \qquad o_{j-1} \le p < o_j, \tag{S8}$$

其中，与主论文中的公式（8）一样，$\mathbf{X}$ 是分段的显式拼接。为评估内存影响，令 $\mathcal{L}_t$ 表示在步骤 t 处实时存在的不同底层分配。指向共享分配的视图只计数一次。我们首先对所有实时分配取峰值，然后更改单个阶段的分配，

$$M_w = \max_t \sum_{a \in \mathcal{L}_t} \mathrm{bytes}(a), \tag{S9}$$

$$M_{\mathrm{before}} = \max(A, C), \qquad M_{\mathrm{after}} = \max(A - d_{\mathrm{save}}, C). \tag{S10}$$

在公式（S10）中，变更阶段的原始分配和净节省字节数分别为 A 和 d_{save}。公式（S9）中的最大值是对每一步求取的，而其他阶段最多使用

表 S19. 适配器与 PTQ4WAM 重放时间之差（秒）。

Task	Mean	SD	95%置信区间	节省（%）	Holm p	A p95	W p95
Long	0.985417	0.045566	[0.956465, 1.014368]	13.4300	1.19269×10^{-15}	7.458	6.495
Push-L	0.411417	0.043829	[0.383569, 0.439264]	11.3950	2.77344×10^{-12}	3.748	3.245
Push-D	2.257917	0.145838	[2.165255, 2.350578]	10.7942	3.49990×10^{-14}	21.095	18.967
Windows	6.427083	0.497172	[6.111195, 6.742972]	11.1689	1.68245×10^{-13}	58.405	51.779

逐点95%区间和 Holm校正配对检验使用 12 次重放。第95百分位是经验最近秩值。

表 S20。四条固定地图路线的时间和峰值分配。

任务 / 指标	q_{00}	q_{10}	q_{01}	q_{11}
Long / 时间（秒）	8.071	7.221	7.096	6.352
Long / 已分配 MB	7681	7103	7390	6846
Push-L / 时间（秒）	3.958	3.653	3.489	3.199
Push-L / 已分配 MB	197.2	173.8	184.5	161.3
Push-D / 时间（秒）	22.170	20.801	20.116	18.660
Push-D / 已分配 MB	1723	1327	1218	936
窗口 / 时间（秒）	61.910	57.406	55.982	51.118
窗口 / 已分配 MB	2810	2409	2538	2206

二进制下标分别表示 FGR 和 DHR。时间四舍五入至毫秒。分配量保留了各工作负载记录的精度。四条路径在每个源上均产生了相同的任务结果。

表 S21. Long 上的一次性准备阶段（秒）。

阶段	时间（秒）
加载	11.7
校准	9.4
提案	1.3
打包	18.9
任务反馈	7211.6
路径分析	16.7
编译/调优	37.8
失败尝试/修复	2.1
总计	7309.5

表 S22。Long 上的准备和预热重放（秒）。

路径	准备	预热重放
BF16-opt	48.6	10.326000
Packed-same-map	7275.0	8.071000
编译适配器	7317.1	7.337417
PTQ4WAM	7309.5	6.352000

C 字节。附录J.7还讨论了由多个张量共享的存储空间，以及影响多个阶段的变更。

D. 精度选择与鲁棒性

D.1. 代理准则

我们依据先前关于如何舍入权重和校准层的工作来评估重构 [20, 28]。曲率加权沿用了将二阶灵敏度用于比特分配的方法 [24, 32]。其他训练后量化方法通过重新缩放或旋转表示来限制离群值的影响 [2, 17, 18, 23, 29]。这两个控制项共享任务选择器的激活格式、量化器和候选映射。更改激活组或添加低秩离群路径 [14] 会改变这三个选择器所使用的模型。

设 $\mathcal{D}_{\text{cal}}$ 为校准输入。对于每个 $x \in \mathcal{D}_{\text{cal}}$，设$\widehat{\mathbf{h}}_w(x)$和$\mathbf{h}_{w,c}(x)$分别为浮点模型与候选项 c中同一选定模型组件的展平输出。重构选择器在待比较的映射集合上最小化以下误差，

$$E_R(c) = \frac{1}{|\mathcal{D}_{\text{cal}}|} \sum_{x \in \mathcal{D}_{\text{cal}}} \frac{\|\mathbf{h}_w(x) - \widehat{\mathbf{h}}_{w,c}(x)\|_2^2}{\|\mathbf{h}_w(x)\|_2^2 + \varepsilon_{\text{rec}}}, \qquad c_R = \arg\min_{c \in \mathcal{C}_w} E_R(c), \tag{S11}$$

其中 $\varepsilon_{\text{rec}} = 10^{-8}$ 用于防止除以零。对于海森选择器，我们根据每个映射的加权误差进行排序，

$$E_H(c) = \sum_{i=1}^{N_w} \operatorname{Tr}(\mathbf{H}_i) \|\mathbf{W}_i - \widehat{\mathbf{W}}_i(c)\|_F^2, \qquad c_H = \arg\min_{c \in \mathcal{C}_w} E_H(c). \tag{S12}$$

这里，$\mathbf{W}_i$ 和$\mathbf{W}_i(c)$分别为原始权重和重构权重，$\mathbf{H}_i \in \mathbb{R}^{d_i \times d_i}$ 是广义高斯-牛顿校准曲率近似$\mathbf{J}_i^\top \mathbf{J}_i$，其中$\mathbf{J}_i$ 是校准输出相对于向量化权重的雅可比矩阵，且 $d_i = \dim \operatorname{vec}(\mathbf{W}_i)$。编码器、预测器以及动作/排序输出的重构分别使用各组件归一化平方误差的权重(0.25, 0.25,0.50)。对角基线使用$0.5 \sum_j h_j (\Delta w_j)^2$：利用32个Rademacher探测向量$\mathbf{z}$ 计算曲率矩阵$\mathbf{H}$ (随机种子20261115)，逐组件计算$\mathbf{h}$ =max(0,mean($\mathbf{z} \odot \mathbf{Hz}$))，以及Tr($\mathbf{H}_i$) $=\sum_j h_j$，并使用数值稳定项 10^{-8}。这些准则对映射进行排序。GPTQ采用单独的权重更新规则。精度搜索中排除的单元，其对应项为常数，因此予以省略。所有选择器共用候选项、校准源、调优预算和量化设置。我们将计算估计值以及收集任务结果所需的时间计入各选择器的成本。

D.2. 量化与存储预算

WxAy 表示 x 位权重和 y 位激活值。受控研究采用 W4/W8 权重和 A8 激活值，其中

表 S23。全部 15 个工作负载的算子覆盖率。

Task	FGR 命中	FGR 回退	FGR 符合条件	DHR 命中	DHR 回退	DHR 符合条件
空间	120	8	128	4	2	6
物体	120	8	128	4	2	6
Goal	120	8	128	4	2	6
Long	120	8	128	4	2	6
Wall-L	36	4	40	3	1	4
Push-L	36	4	40	3	1	4
TwoRoom	36	4	40	3	1	4
Cube	36	4	40	3	1	4
Reacher	36	4	40	3	1	4
Wall-D	22	4	26	3	1	4
PointMaze	22	4	26	3	1	4
Push-D	22	4	26	3	1	4
Meta-World Reach	24	4	28	3	1	4
排序	87	9	96	5	3	8
Windows	86	10	96	5	3	8

符合条件的站点数等于命中数加回退数。详细归因涵盖64次运行研究中的五种工作负载，以及独立12区块研究中的排序。

表S24。Fast-WAM Long上的候选池大小。

候选数量 K	DEV成功次数	最终问答 (%)	准备(s)
12	362/384	94.19	81.7
48	368/384	94.36	271.6
96	370/384	94.29	518.9
192	373/384	94.33	1017.4

表 S25. 对组件名称和提议规则的敏感性。

提议变体	最终 Q (%)
智能体/12 个候选项	94.11
匿名化单元名称	94.09
不使用首选手动 W8 单元	94.02
确定性启发式方法	93.96

表 S26。Long 上的校准输入（成功数/768）。

区块	测量	打乱	移除	启发式
1	727	708	711	713
2	714	698	704	710
3	730	710	715	718
4	721	702	709	712
5	719	707	705	711
6	728	711	714	716
7	709	698	702	712
8	724	706	711	711
9	716	701	708	713
10	733	715	717	719
11	722	707	709	713
12	717	699	706	708
均值 Q (%)	93.9670	91.8186	92.3503	92.8385

Windows 的受保护 A16 操作。表格 S28 给出了每个量化参数对应的序列化字节数。每个候选项都满足相对预算容差 10^{-3}。已登记的 48 映射池的相对预算误差为零。受控研究的分组遵循该表。表格S29 另行记录了选定的部署惯例。我们将舍入到最近值（RTN）量化与 GPTQ 进行比较，后者 [8]，在量化过程中利用曲率更新权重。

D.3. 选择稳定性

在 96 次重复中，12 映射池保持固定，每次重复均使用独立的校准、开发和最终数据源。表格 S30 报告了 96 个区块的平均配对差。

六次规划任务共用一个成功指标。所有八个工作负载的均值还包括排序准确率和正窗口率，因此它描述的是一组端点。每个估计使用 96 个区块、256 个最终数据源，每个数据源用于规划和排序的三个结果，以及每条 Windows 轨迹的四个窗口。表格 S31给出了这些重复比较的准备成本。

D.4. 其他量化比较

原生设置比较使用每个工作负载 48 个区块，每个区块包含 512 个独立源簇。每个工作负载有 24,576 个唯一簇，且各区块之间没有重复标识符。主论文表 1 中的部署结果，以及基于重复执行次数得出的估计，均来自各自采用不同源设计的研究。

表 S32中的原生精度比较使用了与主论文表 1 中部署结果不同的源队列。重放打包映射可以复现 PT Q4WAM 的结果。对于六个全 W4 工作负载，原生 W4 路径同样可以复现这些结果。主论文表 3 中的固定池比较通过将每条规则应用于同一个候选池来比较这些规则。

表 S27. 校准输入对候选名单的配对效应。

对照组	Mean	SD	95%置信区间	Holm p
打乱	2.1484	0.3662	[1.9158, 2.3811]	1.35×10^{-9}
移除	1.6168	0.3825	[1.3737, 1.8598]	2.94×10^{-8}
确定性启发式方法	1.1285	0.6670	[0.7047, 1.5523]	1.09×10^{-4}

差异和标准差以百分点表示。配对区间为逐点区间（$df = 11$）。Holm校正涵盖三项检验。

表 S28. 固定预算下每个符合条件权重的序列化字节数。

工作负载	激活	组	池范围	重建	海森矩阵	Task
Long	A8	32	0.756221–0.756221	0.7562	0.7562	0.7562
Wall-L	A8	32	0.759621–0.759621	0.7596	0.7596	0.7596
Push-L	A8	32	0.650825–0.650825	0.6508	0.6508	0.6508
TwoRoom	A8	32	0.766565–0.766565	0.7666	0.7666	0.7666
Cube	A8	32	0.625362–0.625362	0.6254	0.6254	0.6254
Push-D	A8	32	0.748589–0.748589	0.7486	0.7486	0.7486
排序	A8	32	0.770604–0.770604	0.7706	0.7706	0.7706
Windows	A8/A16 受保护	32	0.793653–0.793653	0.7937	0.7937	0.7937

全部48个映射 满足精确的字节预算。候选池端点精确到小数点后六位一致。选择器特定条目显示到小数点后四位 .

表 S29. 部署中使用的权重和激活格式。

Path	缩放因子、编码和校准
对称W4	分组 $G = 32$， $s = \max \lvert w \rvert / 7$，缩放因子为 1 对于全零分组，零点为 0。将并列值舍入到偶数，截断到 $[-8, 7]$，并先打包低半字节。
W8 权重	对称缩放因子 $s = \max \lvert w \rvert / 127$，舍入并列到偶数并截断到 $[-128, 127]$。
受控激活值	A8 激活值，受保护的 A16 运算适用于 Windows。
原生激活值	A16 激活值在原生量化路径之间共享。
存储空间和保护	有效载荷、FP16 缩放因子、受保护张量、填充，以及头部计入存储空间；归一化，Softmax、嵌入和头部运算仍然受到保护。

表 S30。96 个块上的成对选择器差异（百分点）。

工作负载	任务–重建	任务–海森矩阵	任务–对角线
Long	3.83	3.10	1.56
Wall-L	3.81	3.01	1.62
Push-L	3.58	2.85	1.44
TwoRoom	3.57	2.63	1.56
Cube	3.54	2.65	1.52
Push-D	3.63	2.74	1.41
排序†	3.83	3.14	1.43
Windows†	3.50	2.80	1.49
六次规划	3.66	2.83	1.52
全部八个端点	3.66	2.87	1.50

D.5. 匹配的 LeWM 墙体审计

每一对都共享检查点、规划器、留出的初始状态、停止规则、预处理和评分脚本。主论文表2给出了干预结果

表 S31. 96 个块的准备成本（秒）。

工作负载	Agent-12	Broad-48	缩减
Long	79.4	264.0	69.9%
Wall-L	1004.3	3960.1	74.6%
Push-L	125.8	438.1	71.3%
TwoRoom	520.7	2011.3	74.1%
Cube	414.5	1581.3	73.8%
Push-D	1892.5	7471.7	74.7%
排序†	1297.9	5078.5	74.4%
Windows†	5987.9	23845.1	74.9%
总计	11322.9	44650.1	74.6%

在 48 个配对区块中，每个区块包含 256 个状态。关闭 FGR 后，同一冻结映射的标准化特征误差为 0.0225，参考动作一致率为 93.4%，任务成功率为 94.25%。单独的 12 对分析显示，相较于 FP32 增益为 16.210938 个百分点（95%置信区间 [14.583438, 17.838437]）。调整后的 $p = 0.001953125$ 包含了对我们一并指定的四项检验进行的校正。

表S33 报告了每项条件变更对应的结果。干预比较对两个模型采用相同的特征归一化，并通过更高预算规划或环境标签获取参考动作。

获得等量开发反馈的 LeWM Wall。 独立的： F 协议使用相同的 LeWM 检查点，并为 FP32 规划器调优和 PTQ4WAM 选择各分配 12 个开发配置的预算。两种方法的最终父节点相互匹配，且不用于调优。12 个互不重叠的源区块各包含 256 个最终状态，每个状态对应一个结果。推断使用这 12 个配对区块差异。这种等量反馈对照组与主论文表2中的匹配审计相互独立。表 S34 列出

表 S32. 各原生精度下的任务质量（%）。

Task	FP32	BF16-opt	W4-native	W8-native	Same-map	PTQ4WAM
Fast-WAM 空间	97.93	97.87	96.52	97.71	96.52	96.52
Fast-WAM Object	99.61	99.58	99.39	99.47	99.39	99.39
Fast-WAM 目标	97.15	97.06	94.42	95.91	96.58	96.58
Fast-WAM Long	95.40	95.21	90.23	92.24	93.56	93.56
LeWM Wall	78.03	77.94	89.88	92.07	94.25	94.25
LeWM Push-T	91.24	91.09	87.78	89.62	90.18	90.18
LeWM TwoRoom	86.94	86.81	82.93	85.16	86.17	86.17
LeWM Cube	78.16	78.02	79.66	81.84	84.08	84.08
LeWM Reacher	94.04	93.91	92.12	93.76	92.12	92.12
DINO-WM Wall	94.11	93.92	98.16	95.90	98.16	98.16
DINO-WM PointMaze	99.83	99.80	99.83	99.76	99.83	99.83
DINO-WM Push-T	92.07	91.88	88.66	91.81	92.07	92.07
DINO-WM Meta-World Reach	90.03	89.90	88.14	89.52	88.14	88.14
Franka Ranking	93.45	93.31	87.63	90.04	91.22	91.22
RoboCasa Windows	35.86	35.61	33.56	36.13	37.71	37.71

表S33。Wall 上的控制成功率和严重失败率（%）。

条件	FP32 Q	PTQ4WAM Q	FP32 失败	PTQ4WAM 失败
新状态	77.64	92.97	7.10	2.80
未见目标	75.83	90.41	8.40	3.50
时域 $\times 1.5$	70.56	85.88	11.60	5.20

了每个配对区块的成功次数，表 S35 则量化了两种选择程序之间的差异。

表 S34. LeWM Wall 上的等反馈成功次数（共 256 次）。

区块	FP32 等反馈	PTQ4WAM
1	211	239
2	214	242
3	207	240
4	219	244
5	209	238
6	216	241
7	213	245
8	210	237
9	220	243
10	206	239
11	217	246
12	212	240
均值 Q(%)	83.1380	94.2057

表 S35. 等反馈选择对 Wall 的配对效应。

均值（百分点）	标准差（百分点）	95%置信区间（百分点）	p
11.0677	1.2390	[10.2805, 11.8549]	4.76×10^{-12}

该效应为 PTQ4WAM 减去 FP32。逐点区间和单个双侧配对 t 检验使用相同的 11 个自由度。

E. 候选池限制与准备成本

E.1. 智能体配置与信息边界

可选智能体会限制生成器可生成的精度分配。它接收架构元数据、校准统计数据、存储预算和量化约束。智能体看不到开发集或最终集的源数据或得分。生成器使用固定规则检查重复映射、预算容差、激活与分组规则、受保护操作以及固定顺序的平局裁决。表 S36 说明了提案模型，以及其响应的抽样方式。

智能体仅使用校准输入。表 S37 比较了由此生成的候选短名单与广泛候选池。接口返回两个空列表：必需_w8=[] 和优选_w4=[]。它还返回 $n_4^{\min} = n_4^{\max} = N_4$，最小汉明距离为 2，表 S38中列出了特定任务的候选 ID。记录的 Long 响应拒绝了 C48，并以 C44 替代。Wall-L 响应则将 C48 替换为 C46。拒绝和修复后的响应均保留其协议、候选池、任务和块标识符。每个映射的 ID 均合法，符合字节预算，且在列表中仅出现一次。修复上限为 2，未使用回退。两次修复失败会触发固定的校准启发式方法。表S39报告了选定映射位集和 SHA-256 摘要。重叠情况和验证记录准确显示了哪些映射进入了各个列表。

表 S36. 提案模型和生成器设置。

字段	设置
输入	校准注册表（校准注册表）， 架构元数据、48 个可行映射标识符， 精确的序列化预算和约束， 不包括开发和最终结果。
模型	Qwen3-8B。
信息	智能体和敏感性路径共享校准 信息。
采样	温度 0.6，top-p 0.9，随机种子 20261115。
输出	一个 JavaScript 对象表示法（JSON）物体 包含协议_id、候选池_id、任务、区块， 状态，必需_w8，首选_w4， n4_最小值，n4_最大值，最小_汉明距离，以及 12 候选标识符。_
验证	候选池 48，子集 12，组 32，最小值 汉明距离 2，搜索上限 100,000，至多 两次修复，相对字节预算容差为 0.001。 重新计数精确的序列化字节数，不可行或 重复映射将被拒绝，规范 词典序决胜。

E.2. 候选池审计与参考结果

在记录的 Long 和 Wall-L 区块中，Agent-12 完全由 Broad-48 中的映射组成。开发评分为两个 Long 候选池均选出 C20，为两个 Wall-L 候选池均选出 C34。表S37 报告了它们的池重叠情况。在八种工作负载和每种工作负载的 96 个区块中，共检查了全部 768 个 Agent 候选池。Agent/Broad 重叠的最小值为 12/12，重复地图数的最大值为零，相对预算误差的最大值为 0，且提议器不会接收任何开发或最终得分。Fixed-12 使用预设列表。Sensitivity-12 仅根据校准测量对映射进行排名。

规划比较按每个最终来源的三次执行分组。来自同一 Ranking 回合的比较保持在一起，如同来自同一 Windows 轨迹的窗口。

E.3. 重复提案与敏感性比较

表S40 比较了八种工作负载下的五种提议规则。其中两种工作负载以匕首符号标记，其结果来自录制数据。六任务均值仅对按规划或控制成功率评分的任务取平均。

E.4. 离线成本

在八种工作负载中，穷举搜索对每个区块评估 384 个映射，每个候选名单评估 96 个。准备账本计入提案推理、候选构建和任务评估。表 S31 列出了工作负载级成本，表 S41 则分别列出各阶段。表 S42 列出了计算各代理指标以及运行原始任务进行比较所需的时间。

将候选池限制为单次提案遍历，以减少控制器评估次数。任务评估成本是迹Hessian代理的 20.1953 倍，也是对角Hessian代理的 15.7974 倍。这些准备成本不计入主论文的运行时间比较。主论文的公式 (6) 统计每个阶段的新访问次数。准备总耗时是对每个工作负载取一个均值之和，而不是全部 96 个区块持续时间之和。附录B 将冷启动时间单独列出，每个阶段只计费一次。

E.5. 等反馈分阶段搜索（−CB）

主要提案比较固定了测试映射的数量。这里保留 PTQ4WAM 的初始候选项，并改变用于排序的开发源数量。只有排名靠前的映射会进入后续阶段，每个新增源都会运行至通常的停止点。

Long -CB 实验使用 12 个配对的 -JR Long 源区块，并共享其观测。PTQ4WAM-阶段和 BO-stage 共享一个包含 $12\to4\to2$ 个候选计划、$16\to64\to128$ 个累计开发父项的候选计划。只有新评估的源才会增加访问次数，因此总计为

$$12(16)+4(64-16)+2(128-64)=512, \qquad \text{(S13)}$$

相比 $12(128)=1536$ 对于 PTQ4WAM-full，缩减幅度为 66.6667%。所有访问均覆盖完整任务时域（每个 Long 父项执行三次：1536 次而非 4608 次源内部结果）。较早阶段缓存的结果会被重复使用，选择器无法获取任何未来阶段的结果。BO-stage 从种子选出的四个可行配对开始，并依据已完成的 16 个父项 DEV 结果再提出八个配对。随后，两种方法均保留累计 DEV 分数最高的四个和两个配对，不再提出新配对。BO-stage 的采集模型使用附录 B.2中的受约束贝叶斯优化信息边界。完整 CAL 性能分析、候选项提议、打包、反馈、编译/调优、加载、失败尝试和调度均计入成本。在全部 12 次运行中，两种阶段方法均满足共用的 4000 秒完整准备时间上限。完整评估为阶段方法提供了预算外的质量和成本参考值。

表S43 将分阶段搜索与其全预算对照组进行了比较。表 S44 列出了成对的成功率和准备时间，表 S45 详细列出了首次运行的成本。

最终任务得分在 PTQ4WAM- stage 下为 94.9436%，在 BO-stage 下为 92.8711%。配对增益为 2.0725 ± 0.8244 pp（样本标准差），Bonferroni 双重同时 95%置信区间为 $[1.4554, 2.6896]$ pp，Holm 双重校正为 $p=2.89\times10^{-6}$。相对于完整评估，平均准备时间节省为 57.7555%。配对准备时间节省的同时置信区间为 $[4721.8831, 4836.8669]$ s，Holm 双重校正为 $p=5.37\times10^{-21}$。这是预先指定的两项主要比较。百分比节省量是均值之比。每项计时比较都包含路径性能分析的成本。

表 S37. 首块池重叠和选定映射。

工作负载	智能体∩广泛	固定∩智能体	智能体∩敏感	广泛选择	智能体选择
Long	12/12	4	5	C20	C20
Wall-L	12/12	2	5	C34	C34

表 S38. 按提议规则列出的第一块候选标识符。

Task	Pool	候选标识符
Long	智能体	C00、C03、C06、C09、C12、C16、C20、C25、C29、C33、C38, C44
Long	敏感性	C00、C06、C12、C20、C29、C13、C17、C22、C31、C36、C42、C47
Wall-L	智能体	C02, C08, C14, C19, C23, C27, C30, C34, C37, C40, C43, C46
Wall-L	敏感性	C02, C14, C23, C34, C43, C01, C11, C18, C24, C32, C39, C47

Fixed-12 在两个任务上均使用 C00–C11。每个候选名单都包含其对应的 48 项映射候选池中的十二个不同候选项。

F. 部署记录与存储空间构成

F.1. 计数与汇总统计

表 S46 给出了质量分子、分母和符合条件的模块数量。对于每个工作负载 w 我们计算相对于其 FP32 任务得分的恢复率，计算方式如下：

$$R_w = 100\,\frac{Q_w^{\mathrm{PTQ4WAM}}}{Q_w^{\mathrm{FP32}}}.$$

我们根据各自的工作负载计数计算每个比率，并取 15 个比率的中位数。量化后的任务得分高于 100% 表示任务得分有所提升。对于计时和内存，我们使用各工作负载的部署测量值。表S47汇总了这些比率在不同工作负载下的情况。表S48记录了另外七个工作负载上可用的精度分配数量。

这些汇总涵盖了原生工作负载，而各任务测量的内容各不相同。若先将所有执行时间相加再计算比值，则长工作负载会获得更高权重。质量方向记录的是观察到的计数。对于质量下降的九项任务，可接受损失取决于具体应用。

F.2. 精度组成与制品大小

表 S49给出了重放时间，以及每个已部署映射中以四位表示的权重占比。

当量化单元的大小不同时，模块占比会与参数占比不同。W4 载荷比例为$(P_4/2)/(P_4/2 + P_8 + 2P_{16} + 4P_{32})$，其中 P_b 统计以 b 位存储的参数，而记录账本共有 $P_{32} = 0$。缩放因子、零点、填充、偏置和容器元数据均单独计数。表 S50列出了以十进制兆字节表示的序列化记录。Wall-L 和 TwoRoom 的序列化大小分别为 10.753024 MB 和 10.851328 MB。附录 L以整字节为单位统计相同的记录。

G. 执行协议与系统检查

G.1. 硬件、软件与计时

我们的主要设备是 NVIDIA GeForce RTX 4090。驱动程序报告的驱动可见内存为 24,576 MiB（24 GiB）。编译器融合和低位宽内核库提供了多种避免中间写入的方法 [4, 5, 22]。表 S51 列出了测量中使用的执行协议。Triton 内核在计时前编译。主要测量路径不使用`torch.compile`。适配器对照的编译在计时调用开始前完成。

设置完成后，我们将设备同步，以便对固定工作量进行计时。模型加载、权重打包和 GPU 内核编译均计入准备阶段。主论文中的表 1 报告了 GPU 重放，其任务质量数据使用了独立的来源队列。附录 I 给出了完整模拟器批次的时间。五工作负载模块研究采用 64 次配对运行，随机安排路径顺序并设置同步边界，其中 q_{00} 两个模块均关闭，q_{11} 两个模块均启用。主要时间变化为100($T_{\mathrm{intervention}}/T_{\mathrm{control}} - 1$)，即 64 次配对运行均值之比。配对加速比均值是另一项独立统计量。根据任务结果进行的提前停止已禁用。每项比较中的工作量均保持固定。每次计时调用均遵循相应工作负载的流程。

G.2. 适用性

表S52 列出了配对模块记录所涵盖的工作负载和模块配置。DHR 要求下一个算子直接从独立的上下文张量读取数据。要同时使用两个模块，必须使用量化算子，并且有一个阶段读取存储的视觉历史。

G.3. 模块关闭值与内存核算

表 S53衡量一次启用一个执行模块时，时间和已分配内存的变化。

表 S54 报告了从 q_{00}（两个模块均关闭）到 q_{11}（联合）的 64 次配对固定映射运行。表S53中的逐次运行模块比较，其端点关系为：

$$T_{\mathrm{P},i} = \frac{T_{\mathrm{Q},i}}{1 + \Delta T_{\mathrm{module},i}}, \qquad M_{\mathrm{P},i} = \frac{M_{\mathrm{Q},i}}{1 - R_{\mathrm{module},i}}. \tag{S14}$$

方程（S14）中的联合端点是 64 次配对运行的均值。时间单位为秒，已分配内存单位为十进制 MB。表S55 记录了各跟踪阶段的实时分配量，单位为 MiB（2^{20} 字节）。表S56 报告了全部四条路径的端点。After 等于 Before

表 S39。规范选定位集和 SHA-256 摘要。

选定映射	符合条件的单元	W4/W8 W4 位集（十六进制）和 SHA-256
:S:Long:01:C20	614	369/245位集：c896afecd46ffcdce13ffac3f1f9d9f7a4de96ee74b9f54f4a27c6e4dfaf566ff0f5351b55a5b81b7abde0eb5fe9b727b8b2e9776be4aa5febf716db27a8b5e210bfa7eae77e39a36cd75550f8 *SHA-256*：74d0f77c6bea9d523d624fc1c72eac204314f773272b371c6d369ce4c4fa8310
:S:Wall-L:01:C34	108	72/36 位集：fdeccb8f67bf74e7bf9be89771b0 *SHA-256*：a170dd39cba94f24559d3ea3d7bb37187d34536d78874be7e39bd87d7c1d03d1

按顺序读取十六进制分段，从最高有效位开始。1 表示 W4。末尾的零用于填充前 N 个有效位。SHA-256 对有效 ASCII 位串进行哈希，不包含换行符。

表 S40. 按工作负载和提议规则划分的留出集质量（%）。

工作负载	Agent-12	随机-12	Fixed-12	Broad-48	Sens.-12
Long	94.48	93.21	93.67	94.55	93.89
Wall-L	95.01	93.91	94.36	95.06	94.52
Push-L	91.32	89.94	90.37	91.40	90.62
TwoRoom	86.91	85.69	86.18	86.95	86.40
Cube	84.93	83.57	84.01	85.01	84.33
Push-D	92.94	91.88	92.29	93.00	92.46
排序†	92.18	91.09	91.46	92.22	91.65
Windows†	38.48	36.93	37.59	38.57	37.92
六次规划	90.93	89.70	90.15	91.00	90.37

表 S41. 八个工作负载的准备成本（秒）。

Pool	共享提案	构建/打包	开发评估	总计
Broad-48	203.4	0.0	750.3	43696.4 44650.1
Random-12	203.4	0.6	145.2	10924.4 11273.6
Fixed-12	203.4	0.0	145.2	10924.4 11273.0
Agent-12	203.4	41.8	153.3	10924.4 11322.9
Sensitivity-12	203.4	4.2	145.2	10924.4 11277.2

表 S42. 任务和代理指标评估成本（秒）。

方法	时间（秒）
重构	431.6
迹海森矩阵	558.2
多输出重构	652.9
对角海森矩阵	713.6
任务评估	11273.0

表 S43。Long 上的分阶段搜索，12 次匹配运行。

Arm	访问次数	质量（%）	准备均值（秒）	准备标准差（秒）
PTQ4WAM-full	1,536	95.1714	8275.1917	93.7992
PTQ4WAM-stage	512	94.9436	3495.8167	58.9840
BO-stage	512	92.8711	3503.9083	63.9149

W-full 重用了完整搜索研究中的 Long 臂。匹配运行提供了质量和准备成本数据。

减去按显示精度计算的移除量，再加上新增量。这些局部活动集和完整运行的单次轨迹峰值，不同于以十进制 MB 为单位、对 64 次配对运行取的 q_{00}–q_{11} 均值。计数包括新工作区以及必须保留更长时间的任何存储空间。

V-JEPA 2-AC 的固定映射执行归因。独立的排序协议 :R 在 RTX 4090 上比较 q_{00}、 q_{10}、q_{01}、 q_{11} 以及一个调优后的同映射适配器，使用 W4/W8-A16 和 32 的组大小。在每个区块内，全部五条路径使用相同的量化权重、输入、形状和任务设置。路径顺序通过随机种子 314159 随机化。12 个互不重叠的区块各包含 15 个完整回合，每回合进行三次比较（总计 180 集和 540 次比较）。不同区块的输入可能不同。时间涵盖一个区块中的全部 45 次比较。内存为峰值已分配 MB（10^6 字节）。所有路径共享表 S57 中每个区块的正确计数，因此平均质量为 91.6667%，且 $\Delta Q = 0$ 个百分点。推断采用配对区块（$n = 12$）；八项时间/内存对比构成一个 Holm 检验族，并使用逐点、非同时 95% t 区间（$df = 11$）。此 W4/W8-A16/$G = 32$ 对照组与 A8/$G = 32$ 和 W4A16/$G = 128$ 比较相互独立。表 S58 列出峰值分配，表 S59 报告重放时间和内存的配对差值。在五条路径中，组合路径的平均时间和分配峰值最低。与调优后的同映射适配器相比，其时间减少 6.515%，分配峰值减少 7.209%。八项对比中，经过 Holm 校正的最大 p 为 0.000116。描述性交互效应 $Y_{11} - Y_{10} - Y_{01} + Y_{00}$ 在时间方面为 -0.4058 秒，在分配峰值方面为 22.225 MB，在质量方面为 0 个百分点。

表S55 中的单次执行轨迹给出了用于构建主论文图 4 的分配量。工作区占用了通过直接读取分段而释放的部分存储空间。第 J.7节说明了为什么一旦由不同阶段设置峰值，缩小复制规模就不再产生额外影响。

G.4. 内存诊断与设备配置

已分配内存统计活动张量和工作区。保留内存包括缓存的分配器块。进程内存还可能包括运行时和库状态。表 S56 报告了从 q_{00} 到 q_{11}期间 64 次成对固定映射运行的完整运行分配峰值内存均值，单位为 MB（10^6 字节）。这些值与表 S55中标注的单次执行 MiB 轨迹相互独立。主论文的表 1 报告了我们所部署模型的分数和资源使用情况。

表S60 记录了 24 GiB RTX 4090 的设备配置。

表 S44. Paired Long 的成功次数和准备时间（秒）。

Run	W-full	W-stage	BO-stage	完整准备	W-stage 准备	BO-stage 准备	性能分析
01	737	735	713	8233.9	3469.4	3442.7	973.4
02	722	722	704	8333.8	3501.1	3513.0	986.7
03	730	727	710	8161.1	3454.3	3467.5	961.2
04	750	751	733	8388.4	3449.6	3464.6	994.8
05	736	732	717	8261.3	3531.5	3498.3	979.6
06	732	731	711	8334.7	3532.7	3533.5	1002.3
07	721	721	706	8140.1	3464.1	3478.4	968.1
08	729	726	728	8360.3	3559.4	3611.1	990.5
09	737	735	714	8218.9	3525.3	3551.4	981.7
10	726	725	705	8253.8	3398.1	3398.2	957.9
11	733	729	716	8432.1	3612.6	3609.4	1009.4
12	718	716	702	8183.9	3451.7	3478.8	965.8

每次运行均有768个最终结果。性能分析费用已计入每项总计。

表 S45. 分阶段搜索的准备阶段，运行 01（秒）。

组件	秒
加载	12.4
校准	9.1
提案	1.5
打包	20.2
反馈	2381.5
性能分析	973.4
编译/调优	36.4
失败/修复	2.7
调度	32.2
总计	3469.4

G.5. 固定工作量执行

固定工作量比较针对每条路径使用 64 次配对运行，且精度映射和工作负载计划相同。表S61 根据配对平均时间计算增量加速比 $S_{\text{material}} = T(q_{00})/T(q_{11})$。

G.6. 数值一致性

每条数值路径均使用相同的代码、缩放因子、零点、精度映射和输入，与参考值进行核对。FP32 输出检查使用最大绝对误差$\leq 10^{-5}$ 和相对$L_2 < 10^{-5}$。对于 16 位浮点（FP16）输出，最大绝对误差为 $\leq 3 \times 10^{-3}$，相对 $L_2 < 10^{-3}$。表S62报告最大绝对误差和相对L_2 误差。使用图形处理器 FGR/DHR 内核进行的 448 个记录案例共包含 896 次 FP32/FP16 检查，均满足这些容差。对于 DHR， $0/0$ 表示测得的最大绝对误差和相对L_2 误差均为零。三元组 $0/0/0$ 记录了输出一致，且分派和输入验证均无失败。

H. 补充测量研究范围

区块 B–H介绍了模拟器执行、记录数据评估和 GPU 重放方面的测量研究。表S63总结了这些研究所报告的范围。完整的模拟器流水线单独计时。补充材料中的 B–H 表报告了各条件下的比较结果和选定的实现细节。四项统计检验还包括原始配对观测。定量汇总见前文表格和附录K。

I. 补充对照比较

表S64 比较了共享候选池中更强的代理选择器。表S65 测量了各个内核，而表S66 比较了使用调优适配器的完整运行。

调优后的适配器采用编译后的融合尾处理，并在相同映射、任务、形状、数据类型和输入下使用分段上下文消费者，且仅进行校准调优。表 S14 比较 CUTLASS 和 Mar- lin 的实现。它使用 `torch.compile`。编译、调优和 20 次预热运行不计入固定工作量执行计时。表 S65 中的算子级测量值与表 S66 中的固定工作量执行使用不同的时间单位和测量规范。附录 B中的对照组涵盖指定实现和任务感知方法、上下文与组大小的变化，以及冷启动的成本。表 S67报告了输入分布变化时的任务质量。

在原始输入和偏移输入上进行的比较包含 48 个区块，每个区块包含 512 个独立的源簇。对于 Windows，我们将一条轨迹中的全部四个窗口作为一个源簇保留在一起。

表 S68 检验在一个任务上选出的映射应用于使用相同检查点的新任务时的表现。两项比较中，目标重新选择带来的增益都需要新的任务反馈。

Long 分布外评估使用未见过的起始状态和目标，时域为 $1.5\times$。TwoRoom 使用未见过的起点–目标对，通道宽度缩窄 20%。Windows 使用不重叠轨迹，亮度因子为 0.8/1.2，且每张图像有十分之一被遮挡。为了迁移到 Long，我们在 Long 的存储预算（369/614 个 W4 单位）内，依据 Goal 校准和开发

表 S46. 部署结果和四位模块计数。

模型 / 任务	FP32	FINAL	Δ质量（百分点）	W4 / 符合条件
Fast-WAM 空间	490/500	482/500	-1.60	614/614
Fast-WAM Object	498/500	497/500	-0.20	614/614
Fast-WAM 目标	486/500	482/500	-0.80	610/614
Fast-WAM Long	477/500	467/500	-2.00	369/614
LeWM Wall	39/50	47/50	+16.00	72/108
LeWM Push-T	137/150	135/150	-1.33	92/108
LeWM TwoRoom	87/100	86/100	-1.00	72/108
LeWM Cube	39/50	42/50	+6.00	100/108
LeWM Reacher	47/50	46/50	-2.00	108/108
DINO-WM Wall	47/50	49/50	+4.00	66/66
DINO-WM PointMaze	50/50	50/50	0.00	66/66
DINO-WM Push-T	46/50	46/50	0.00	42/66
DINO-WM Meta-World Reach	45/50	44/50	-2.00	72/72
V-JEPA 2-AC 排名†	42/45	41/45	-2.22	160/261
V-JEPA 2-AC Windows†	20/56	21/56	+1.79	144/259

表 S47. 15 个工作负载的描述性汇总。

指标	中位数	算术平均数	几何平均数
工作负载加速比	2.36×	2.84×	2.73×
峰值内存降低幅度	71.46%	73.83%	73.18%
FP32 质量恢复	99.18%	101.56%	101.40%

观测质量：9 项下降、2 项并列、4 项增益。

表 S48。七个额外工作负载上的地图计数。

工作负载	评估的地图 K	W4/符合条件
空间	1	614/614
物体	1	614/614
目标（128 个开发集来源）	12	610/614
Reacher	1	108/108
Wall-D	1	66/66
PointMaze	1	66/66
Meta-World Reach	1	72/72

源选择一张映射，不使用 Long 开发或最终得分。主论文表 1 中 610/614 个 W4 单位的 Goal 部署映射是另一件制品。从 Goal 迁移的映射使用键 :Goal-at-Long-budget。我们将目标任务上选定的映射记为 :Long-reselected。对于这次重新选择，我们在 Long 开发源上为 12 张映射评分，其字节数的相对误差至多为 0.001。Wall-L到Push-L比较使用 72/108 个符合 W4 条件的单元、$G = 32$、A8，以及相同的目标源协议。迁移映射在 Push-L 上的得分比目标重新选择低 2.32 个百分点，在 Long 上低 1.43 个百分点。

表 S69 比较了每组时间和内存限制下的两种单模块路径。所有评估的源在四种实现中都产生了相同的结果。表 S70 记录了模拟器完整批次的用时，表 S71 区分了冷启动准备与热执行。表S72 记录了用时和成功情况针对十二组配对批次中的每一组。

模拟器流水线协议。 500 个回合各运行 180 个环境步数，即每批次 90,000 步。每个动作块包含 12 步，共需 7,500 次模型调用，批大小为 1，上下文为 64，且不提前停止。PTQ4WAM 使用冻结的 -J:Long 胜出映射。模型阶段时间对应 75 次、每次 100 个调用的重放。环境推进、规划/调度和输入/输出分别计入完整流水线。热启动加速比为 $974.250/676.200 = 1.4408\times$。记录的 -Z 非模型项在所显示的精度下相同。仅凭这些值无法区分共享计量与独立阶段计时。单独的 -ZR 协议会独立计时全部四个阶段。计入准备后，对于 N 这样的批次，$T_{\mathrm{BF16}}(N) = 48.6 + 974.25N$ 和$T_{\mathrm{PTQ4WAM}}(N) = 7309.5 + 676.20N$。PTQ4WAM 总耗时首次更短时的批次数为 25。这些测量涵盖模拟控制、基于录制数据的任务，以及固定的图形处理器重放。

独立计时模拟器批次 (–ZR)。 来自 -CB 长程运行 01 的预先声明的 PTQ4WAM-stage 制品在 12 个配对模拟器批次中保持固定，每个批次包含 500 个新的父级回合，且与 -JR/-CB 不重叠。每个回合包含 180 个环境步数，动作块长度为 12（每个批次 7500 次模型调用），批大小为 1，上下文为 64，且不提前停止。BF16 和 PTQ4WAM 分别计量模型执行、环境推进、规划/调度和输入/输出所需的时间。配对运行顺序为 AB、BA、BA、AB、AB、BA、BA、AB、BA、AB、AB、BA。 A 表示 BF16，B 表示 PTQ4WAM。这些完整批次的时间与 100次调用重放的第 95百分位以及单次调用的 -JR 延迟上限均不同。

配对批次节省量为 297.316417 ± 5.372640 秒（样本标准差），双侧 95% 区间为$[293.902804, 300.730030]$ 秒，配对-t $p = 9.76 \times 10^{-21}$

表 S49。部署重放时间和四比特组成。

工作负载	FP32（秒）	PTQ4WAM（秒）	峰值节省率（%）	W4占比（%）
Fast-WAM 空间	143.450	33.398	78.06	100.00
Fast-WAM Object	149.432	38.642	78.06	100.00
Fast-WAM 目标	145.039	33.948	78.00	99.35
Fast-WAM Long	13.166	6.569	71.46	60.10
LeWM Wall	23.126	10.705	63.10	66.67
LeWM Push-T	10.694	3.317	64.69	85.19
LeWM TwoRoom	23.202	10.738	63.10	66.67
LeWM Cube	11.863	4.169	65.23	92.59
LeWM Reacher	11.340	3.069	69.70	100.00
DINO-WM Wall	10.349	5.030	80.34	100.00
DINO-WM PointMaze	14.161	5.992	95.31	100.00
DINO-WM Push-T	41.243	19.918	95.27	63.64
DINO-WM Meta-World Reach	138.758	58.910	65.53	100.00
Franka 排序	37.507	12.236	73.65	61.30
RoboCasa Windows	114.990	53.734	65.97	55.60

主论文中的表 1 给出了相同精度下的质量、峰值分配和加速比。我们测量相较于 FP32 的峰值节省率，并计算符合条件的组件中的 W4占比。

表 S50。权重组成与序列化制品大小。

Task	模块 W4	参数 W4（全部）	W4 载荷	制品（MB）	压缩
LeWM Wall	66.67%	66.8182%	48.8372%	10.753024	5.363299×
LeWM TwoRoom	66.67%	65.4545%	47.3684%	10.851328	5.314712×

表 S51。执行软件和测量设置。

设置	值
预热	20
每个工作负载和路径的配对运行次数	64
块内路径顺序	随机排列
Triton 即时编译	不计入计时
torch.compile	在主要路径上已禁用
英伟达驱动程序 / CUDA 工具包	570.124.06 / 12.8
PyTorch / Triton	2.6.0 / 3.2.0
提前停止	已禁用

表 S52. 配对算子比较所涵盖的工作负载。

工作负载	FGR	DHR	联合	配对运行/区块
Push-L	Yes	Yes	Yes	64
Cube	Yes	Yes	Yes	64
Long	Yes	Yes	Yes	64
Wall-D	Yes	Yes	Yes	64
PointMaze	Yes	Yes	Yes	64
排序（:R）	Yes	Yes	Yes	12

以及精确符号 $p = 0.000488281$。完整流程耗时减少了 30.5552%。我们在表 S71中报告成功次数，作为描述性计数。

令 P_B，P_W 分别表示 BF16 和 PTQ4WAM 的初始准备成本，T_B，T_W 表示……中的平均预热批次时间

表 S53. 固定精度下 FGR 和 DHR 的独立影响。

工作负载	模块	内存 ↓	时间变化
Push-L	FGR	17.10%	-7.41%
Cube	FGR	16.22%	-9.31%
Long	FGR	18.58%	-10.53%
Wall-D	DHR	13.39%	-7.62%
PointMaze	DHR	14.13%	-8.83%

变化量为 64 次配对运行的均值。打包路径不会启用任何模块。负的时间变化表示执行速度更快。

表 S54。64 次配对运行中的时间与峰值的联合变化。

工作负载	时间（秒）	峰值（MB）	时间 ↓	峰值 ↓
Push-L	4.140 → 3.342	223.4 → 166.4	19.28%	25.51%
Cube	5.112 → 4.208	215.4 → 164.0	17.68%	23.86%
Long	8.382 → 6.595	9812.7 → 7111.8	21.32%	27.52%
Wall-D	6.330 → 5.052	1190.4 → 873.8	20.19%	26.60%
PointMaze	7.560 → 6.006	1241.6 → 922.2	20.56%	25.72%

表 S71。首次严格更快的批次数为

$$N_* = \max\left(1, \left\lfloor \frac{P_W - P_B}{\overline{T}_B - \overline{T}_W} \right\rfloor + 1\right). \tag{S15}$$

如果制品已存在，加载需要 43.1 s，其中包括 14.7 s 的读取时间、2.8 s 的验证时间、8.9 s 的 GPU传输时间和 16.7 s 的设置时间。由此得到的首批次在线总耗时，BF16 为 1,021.647167 s，PTQ4WAM 为 718.830750 s，因此加速比为 1.421262×。将加载模型所需的时间计入后，PTQ4WAM 曲线将偏移 43.1 s，并

表 S55。历史分配与全模型峰值（MiB）。

工作负载 / 模块	之前	移除	新增	之后	全局峰值
Wall-D / DHR	1140.1	291.6	72.6	921.1	1140.1 → 988.8
PointMaze / DHR	1191.3	283.6	74.0	981.7	1191.3 → 1022.0

表 S56. 四条固定地图路线的绝对结果。

工作负载 / 指标	q_{00}	q_{10}	q_{01}	q_{11}
Push-L / 时间（秒）	4.1400	3.8332	3.6279	3.3420
Push-L / 峰值 (MB)	223.40	185.20	199.67	166.40
Cube / 时间（秒）	5.1120	4.6361	4.5901	4.2080
Cube / 峰值 (MB)	215.40	180.46	194.38	164.00
Long / 时间（秒）	8.3820	7.4994	7.2974	6.5950
Long / 峰值 (MB)	9812.70	7989.50	8624.38	7111.80
Wall-D / 时间（秒）	6.3300	5.4337	5.8477	5.0520
Wall-D / 峰值（兆字节）	1190.40	1003.03	1031.01	873.80
PointMaze / 时间（秒）	7.5600	6.5303	6.8925	6.0060
PointMaze /峰值（兆字节）	1241.60	1061.82	1066.16	922.20

我们对 64 次配对运行取平均值，并以秒报告时间、以十进制兆字节报告峰值分配内存。

使整数交叉点保持不变。-Z 和 -ZR 的测量值采用不同的制品和计时协议。

J. 形式分析与证明

该分析支持 PTQ4WAM 的两个阶段。对于精度选择，我们在指定源分布下，相对于有限有效池中的最佳映射，对任务得分的期望损失进行界定。对于执行，算子恒等式为固定的一组量化操作数建立等价性，而活跃集记账则将局部分配变化与全局内存峰值联系起来。运行时和分配器预留容量将独立于该分配模型进行评估。表 S73 定义了两组证明中使用的符号。

J.1. 记号与概率空间

假设1（映射的构建方式及源的采样方式）。 令 $\mathscr{F}_w$ 包含所有开发前信息和随机性，包括校准、Agent提案和候选构建。在给定 $\mathscr{F}_w$的条件下，有序候选池和制品是固定的，且开发源簇 $Z_1, \dots, Z_n$ 均独立同分布于 $\mathcal{P}_w$，其中 $n \geq 1$。对于一个源，其任务得分 $v(c, Z)$ 称为其效用，该得分是可测的，且取值于 $[0, 1]$。来自同一源的重复执行或窗口属于同一源簇。选择器最大化经验均值，并按照映射的固定顺序解决并列情况。

取一个概率空间 $(\Omega\ \mathscr{F},\ \mathbb{P})$，其承载校准数据、提案随机性和开发集群，并令 $\mathscr{F}_w \subseteq \mathscr{F}$ 为假设 1中的σ域。所有候选池都是有限合法 W4/W8映射空间的非空有限子集，并且是 $\mathscr{F}_w$的可测函数。候选项制品包括固定的量化统计量和任何打包随机性。在给定 $\mathscr{F}_w$、 Z_j 的条件下，取值于可测源空间 $\mathcal{Z}_w$，且 $v(c,\ \cdot)$: $\mathcal{Z}_w \to [0, 1]$ 固定。为简洁起见， $q(c)$表示 $\mathbb{E}[v(c,\ Z) \mid \mathscr{F}_w]$。该量可随实际校准结果和制品而变化。

Z_j中纳入了源级执行随机性。同一源内，重复执行和不同候选项的效用可能彼此相关。该假设要求不同源簇之间相互独立，即使在给定任何共享随机变量的条件下也是如此。互不重叠的源标识符可避免重叠，但我们还假设每个源都是从其工作负载分布中独立抽取的。

方程（S12）中的海森代理提供了一种按迹加权的比较准则。定理1 不要求该准则与任务效用之间存在任何关系，也不对$\mathbf{H}_i$施加正值性要求，也不将其解释为任务损失。其估计器和数值稳定器的取值属于实现细节，相关报告范围概述于附录N。

J.2. 有限池任务选择

源的数量有限时，映射的选择可能因样本而异。对于固定的有限候选池，集中界可以限制相对于其最佳映射所损失的效用。候选覆盖将在附录J.3中单独讨论。假设1 规定候选项须在开发评分前固定，并按源对重复执行进行分组。在给定开发前信息 $\mathscr{F}_w$的条件下，源簇独立同分布，且效用取值于$[0, 1]$。以 $\mathscr{F}_w$ 为条件，并定义 $q(c) = \mathbb{E}[v(c, Z)]$ 以及$\widehat{q}_n(c) = n^{-1} \sum_{j=1}^{n} v(c, Z_j)$。等源选择使用$\widehat{Q}_w^{\mathrm{dev}} = 100\widehat{q}_n$。由于对两个映射都使用相同的源进行测试，它们的分数可能彼此相关。

定理1 (有限候选池任务选择界)。 在假设 *1*下，令 $c^\star \in \operatorname{argmax}_{c \in \mathcal{C}_w} q(c)$ 表示包含 $K \geq 1$ 个候选项的固定候选池中的 c_T，并令其最大化 $\widehat{q}_n$。对于 $0 < \delta < 1$，以下事件在抽取开发源时发生的概率至少为 $1 - \delta$:

$$q(c^\star) - q(c_T) \leq \min\{1, 2\eta_K\}, \qquad \eta_K = \sqrt{\frac{\log(2K/\delta)}{2n}}. \tag{S16}$$

证明见附录J.4。公式（S16）使用单位区间上的效用。将两边同时乘以 100，即可得到以百分点表示的任务得分界限。

表 S57. 排名重放时间（秒）和正确比较次数。

区块	q_{00}	q_{10}	q_{01}	q_{11}	调优适配器	正确
1	15.0695	13.9707	13.8684	11.2752	12.1573	40
2	15.8061	14.4978	14.3235	12.6349	13.3949	42
3	15.2815	14.2227	14.3196	11.5150	13.0022	41
4	15.7798	14.2866	14.6952	12.6216	13.6068	43
5	16.1835	14.3032	15.5013	12.4960	13.6990	41
6	15.4583	13.9675	14.4721	12.6108	12.9797	39
7	14.9834	13.1922	13.6876	12.0089	13.1214	42
8	14.5952	12.6093	13.2190	11.6876	11.6697	40
9	15.7710	14.5654	14.2514	13.7112	14.5738	41
10	14.8360	13.3197	14.0922	11.4274	13.1235	42
11	15.7315	14.6211	14.3158	12.7117	12.8195	43
12	15.6422	13.6973	14.0064	13.2981	14.1652	41
Mean	15.4282	13.9378	14.2294	12.3332	13.1928	91.6667%

任务计数表示五条路径各自 45 次比较中的正确比较次数。最后一项给出平均质量。

表 S58。排名峰值分配内存（十进制 MB）。

区块	q_{00}	q_{10}	q_{01}	q_{11}	调优适配器
1	1998.4	1688.3	1779.2	1495.6	1560.1
2	2016.5	1770.1	1802.0	1514.6	1637.5
3	2099.0	1783.5	1876.6	1639.9	1798.8
4	1998.8	1689.3	1786.2	1625.1	1727.9
5	2070.7	1807.7	1884.8	1596.4	1747.6
6	2140.1	1854.6	1995.0	1670.8	1796.4
7	2047.9	1737.2	1870.5	1575.4	1715.9
8	2050.8	1704.6	1789.4	1518.7	1579.7
9	2063.7	1758.6	1793.2	1531.9	1696.3
10	2042.7	1732.4	1915.3	1502.9	1692.6
11	2082.2	1764.3	1911.2	1613.1	1712.7
12	2034.5	1678.5	1812.8	1522.3	1602.3
均值	2053.775	1747.425	1851.350	1567.225	1688.983

J.3. 候选收缩与覆盖

公式（S6）展示了提案模型如何将约束传递给映射生成器。设 Θ_w 为预设的合法~提案区域。智能体依据架构元数据、校准统计数据、存储预算以及量化和硬件约束，将该区域限制为 Θ_w。确定性生成器 Γ_w 会返回 12 个候选项，随后由主论文第 3.1 节中的任务选择器对其进行评估。智能体指定 W4 数量目标或窗口、必需或优选的 W8/W4 分配，以及多样性约束。这些条件会在任务评分之前指导候选构建。已注册的 Wall-L 接口没有必需的 W8 单元，并且保留恰好 N_4 个 W4 单位，同时允许十二种不同的映射。确定性生成器 Γ_w 使用固定随机种子，并在激活精度、组大小、受保护单元和硬件约束不变的情况下，强制满足公式（S3）。量化格式和选择规则保持不变。智能体输入不包含开发和最终数据源，也不包含候选项得分。候选池及其排序在开发评估之前即被冻结。模型和提示词、候选池审计、重复运行、敏感性分析以及准备成本见附录 E。

J.3.1. 收缩下的候选覆盖

候选名单所需的任务试验较少，但也可能排除期望效用更高的映射。我们以任务评分前固定的广泛候选池为基准衡量覆盖，并将这种损失与受限集合内的选择误差区分开来。

推论 1（收缩将覆盖与选择区分开来）。 在假设*1*成立的条件下，对于非空的开发前候选池 $\mathcal{C}_w^B$ 和 $\mathcal{C}_w^A$，定义 $q_B^\star = \max_{c\in\mathcal{C}_w^B} q(c)$、$q_A^\star = \max_{c\in\mathcal{C}_w^A} q(c)$，以及 $\rho_A = [q_B^\star - q_A^\star]_+$，其中 $[x]_+ = \max(x,0)$。对于从智能体候选池的 c_T^A 个映射中选出的 K，以至少 $1-\delta$ 的概率，遗憾满足

$$q_B^\star - q(c_T^A) \le \min\{1, \rho_A + 2\eta_K\}. \tag{S17}$$

证明见附录 J.4。此处 ρ_A 是两个候选池最优值之间的正效用差距。较小的K 会降低集中度惩罚，但无法控制覆盖损失。因此，我们针对广泛、随机和校准敏感性候选池评估收缩。分数比较的是完整的冻结搜索路径，其差异也包含选择误差和评估采样变化。它们并不直接估计 ρ_A（附录 J.4）。该分析要求在开发评分前固定智能体的随机性，并要求源簇相互独立。标识符不相交可确保源记录彼此分离。抽样独立性则是额外的假设。

J.4. 完整有限候选池与收缩证明

以下所有陈述均以 $\mathscr{F}_w$ 为条件，除非另有说明。假设 1 在开发评分之前固定候选池，并要求独立的有界源观测。校准数据、架构元数据和量化约束可以为智能体和合法生成器提供信息。开发结果不纳入候选池构建。

表 S59. 排序上的配对时间和内存效应。

对比	Unit	Mean	SD	95%置信区间	减少率（%）	Holm p
FGR ($q_{00}-q_{10}$)	s	1.4904	0.3429	[1.2725，1.7083]	9.660	4.39×10^{-8}
FGR（$q_{00}-q_{10}$）	MB	306.3500	30.5083	[286.9660，325.7340]	14.916	9.31×10^{-12}
DHR（$q_{00}-q_{01}$）	s	1.1988	0.3090	[1.0024，1.3952]	7.770	1.08×10^{-7}
DHR（$q_{00}-q_{01}$）	MB	202.4250	42.8690	[175.1873, 229.6627]	9.856	2.28×10^{-8}
联合（$q_{00}-q_{11}$）	s	3.0950	0.5352	[2.7549, 3.4350]	20.060	3.15×10^{-9}
联合（$q_{00}-q_{11}$）	MB	486.5500	45.2806	[457.7801，515.3199]	23.691	5.08×10^{-12}
调优后的 $-q_{11}$	s	0.8595	0.5115	[0.5346, 1.1845]	6.515	1.16×10^{-4}
调优后的 $-q_{11}$	MB	121.7583	41.1342	[95.6229, 147.8938]	7.209	1.15×10^{-6}

时间和内存增益采用配对差值计算。区间为逐点 95% 区间，p 值采用八项检验的 Holm校正。百分比降幅根据两条路径均值的比值计算。

表 S60. GPU 硬件和软件配置。

字段	值
主要驱动程序报告的型号	RTX 4090
主要驱动程序可见容量（MiB）	24,576
设备配置容量	24 GiB
计算能力	8.9
驱动程序	570.124.06
CUDA / PyTorch / Triton	12.8 / 2.6.0 / 3.2.0

表 S61. 固定工作量重放时间和增量加速比。

工作负载	打包 q_{00}	联合 q_{11}	S_{material}
Push-L	4.140	3.342	1.2388×
Cube	5.112	4.208	1.2148×
Long	8.382	6.595	1.2710×
Wall-D	6.330	5.052	1.2530×
PointMaze	7.560	6.006	1.2587×

引理1（有界源集中）。固定 c 和容差 $\eta>0$。对于此候选项，采样误差满足

$$\mathbb{P}\big(|\widehat{q}_n(c)-q(c)|\ge\eta\mid\mathscr{F}_w\big)\le 2e^{-2n\eta^2}. \tag{S18}$$

证明。为证明公式 (S18)，我们考虑一个随机变量 $V\in[0,1]$，并定义 $f(\lambda)=$日志 $\mathbb{E}\exp(\lambda(V-\mathbb{E}V))$，其中 $\lambda\in\mathbb{R}$。由于该变量有界，我们可以将求导运算移入期望。在指数倾斜分布下，$f''(\lambda)$ 是一个方差。对于任意 $U\in[0,1]$，$\mathrm{Var}(U)\le\mathbb{E}U-(\mathbb{E}U)^2\le 1/4$，因为 $U^2\le U$。因此$f''(\lambda)\le 1/4$，并且由于 $f(0)=f'(0)=0$，泰勒积分公式得出 $f(\lambda)\le\lambda^2/8$，对 λ的任一符号均成立。对于$V_j=v(c,Z_j)$ 和 $\lambda>0$，独立性给出下式。我们对指数应用马尔可夫不等式，以界定上尾概率，

$$\mathbb{P}(\widehat{q}_n(c)-q(c)\ge\eta)\le e^{-\lambda n\eta}\prod_{j=1}^{n}\mathbb{E}e^{\lambda(V_j-q(c))} \tag{S19}$$

$$\le\exp(-\lambda n\eta+n\lambda^2/8). \tag{S20}$$

选择 $\lambda=4\eta$ 得到 $e^{-2n\eta^2}$。将该论证应用于$-\sum_j($ $V_j-q(c)$)，并合并两侧的尾部。 □

定理的完整证明*1*。将引理 1 应用于所有 K 固定候选项，并取并集界，得到

$$\mathbb{P}\left(\max_{c\in\mathcal{C}_w}|\widehat{q}_n(c)-q(c)|\ge\eta\right)\le 2Ke^{-2n\eta^2}. \tag{S21}$$

设 $\eta=\eta_K=\sqrt{\log(2K/\delta)/(2n)}$。在其补事件上，所有映射都满足相同的误差界，

$$\begin{aligned} q(c^\star)-q(c_T)&=[q(c^\star)-\widehat{q}_n(c^\star)]\\ &\quad+[\widehat{q}_n(c^\star)-\widehat{q}_n(c_T)]\\ &\quad+[\widehat{q}_n(c_T)-q(c_T)] &\text{(S22)}\\ &\le\eta_K+0+\eta_K. &\text{(S23)}\end{aligned}$$

中间项因经验最优性而非正，包括并列情形。所有效用都位于 $[0,1]$, 内，因此遗憾也至多为 1。这证明了公式（S16）。允许 K 候选得分向量彼此依赖。将效用界乘以 100，即可得到其百分点形式。 □

评分前固定的随机候选池。对于 $\mathscr{F}_w$ 的几乎所有实现，条件失败概率至多为 δ。取其期望可得，无条件失败概率至多为 δ。因此，该论证允许随机的 Agent 提案，无需对提案历史取并集界，同时保留已实现候选池中的最佳候选作为参考值。如果候选池取决于开发分数，那么之后将其固定也无法恢复上述条件化所需的条件。

推论的完整证明*1*。在给定 $\mathscr{F}_w$的条件下，两个参考池均为固定的。加上并减去 $q_A^\star$:

$$\begin{aligned} q_B^\star-q(c_T^A)&=(q_B^\star-q_A^\star)+(q_A^\star-q(c_T^A))\\ &\le\rho_A+(q_A^\star-q(c_T^A)). \end{aligned} \tag{S24}$$

应用定理1 于 $\mathcal{C}_w^A$，然后使用 $q_B^\star-q(c_T^A)\le 1$。这证明了该界限具有相同的失败概率，且无需假设嵌套。当 $\mathcal{C}_w^A\subseteq\mathcal{C}_w^B$，$\rho_A=q_B^\star-q_A^\star$。对于分别生成的合法池，有符号差可能为负，而正部则保留了该界限。 □

表 S62. 与量化参考值的数值一致性。

工作负载/设计	案例	FP32（绝对/相对）	FP16（绝对/相对）	失败
Push-L / FGR	96	$2.86 \times 10^{-6}/4.17 \times 10^{-7}$	$9.77 \times 10^{-4}/2.14 \times 10^{-4}$	0/0/0
Cube / FGR	96	$3.81 \times 10^{-6}/6.38 \times 10^{-7}$	$1.46 \times 10^{-3}/3.72 \times 10^{-4}$	0/0/0
Long / FGR	64	$7.15 \times 10^{-6}/8.61 \times 10^{-7}$	$1.95 \times 10^{-3}/5.38 \times 10^{-4}$	0/0/0
Wall-D / DHR	96	0/0	0/0	0/0/0
PointMaze / DHR	96	0/0	0/0	0/0/0

相对于量化参考值的最大绝对和相对 L_2 误差。失败项计数包括不匹配、调度错误和验证错误。

表 S63。补充研究的覆盖范围和解读。

区块	记录的比较	可用的解读
字节：部署控制项	源匹配的 FP32、BF16-opt、W4-native、W8-native，同映射执行，以及 PTQ4WAM 比较。	同映射路径可隔离执行变更。测量形状和重复次数是所述输入。
C：候选项来源	每个受控工作负载有 48 个合法映射，固定 12 映射选择器池、选定映射、重叠、字节预算，以及仅用于开发的选择。	表格列出了选定标识符、两个选定映射位集、SHA-256 摘要、打包大小范围和源数据划分。
D：离线成本	提案推理、代理遍次、加载、校准、打包、预热、任务评分、序列化、反馈获取和总搜索时间。	报告的成本描述了其自身的准备协议。路径分析和冷启动成本详见附录 字节。
E：模块消融	模块关闭、仅 FGR、仅 DHR 与联合执行比较。	配对记录涵盖五种支持联合执行的工作负载，每种配置运行 64 每种配置运行次数，以及 12 个区块时的排序，单位为字节。N 协议中对 12 个区块进行排序：R。峰值节省率取决于活动集。
G：迁移与分布偏移	Goal-to-Long 和 Wall-L-to-Push-L 迁移，匹配分布外（OOD）基线，冻结映射下分布偏移，以及在分布偏移后的评估条件。	结果适用于已评估条件和源级传输比较。
H：Wall 机制	FP32、PTQ4WAM、匹配噪声、敏感与随机恢复、关闭 FGR 和精度置换。	48 个配对区块描述了所评估的干预。单独的 :F 对照组同时提供方法、12 次开发评估。因果结论仅限于这些条件。
-J：可行选择	四项任务、48 映射池、匹配的 12 条反馈提案、贝叶斯优化和 Pareto-48、冻结映射最终评估监控。	配对最终计数、同时区间和资源触发回退详见附录字节。
-X/-Z：执行	同地图重放对照以及完整的 Long 模拟器流水线（含准备）。	模型执行、环境推进、规划/调度、输入/输出（I/O）和摊销分别核算。
-JR：完整搜索	四个系列，每个系列进行 12 次独立的 48 映射搜索，使用相同的编译后端、12 对贝叶斯优化和质量优先对照组。	运行级最终计数、约束检查，以及附录中的 Holm-8统计量B.2。
-CB：分阶段反馈 Long	PTQ4WAM-stage 和 BO-stage，每个阶段有 512 次候选父节点访问。	完整准备和任务质量结果，附录E.5。
-ZR：完整流水线	十二个独立计时的配对批次，每批包含 500 个新模拟器回合。	完整阶段计时、源计数、冷启动准备和预构建加载，每个配对批次对应一次观测。

Broad-48 本身使用经验任务选择。若 c_T^B是其选定映射，且 $r_s = q_s^\star - q(c_T^s) \geq 0$适用于 $s \in \{A, B\}$，则

$$q(c_T^B) - q(c_T^A) = (q_B^\star - q_A^\star) + r_A - r_B. \qquad \text{(S25)}$$

除方程（S25）中的各项外，最终面板的差异还包含评估抽样变异。因此，观测到的 Agent–Broad差距比较的是冻结搜索路径，无法直接估计 ρ_A。附录 E审计了 Long 和 Wall-L 中 Agent- 12 是否属于 Broad-48。这种归属关系是候选池的一项属性，需要其最终得分之外的证据。无论一个候选池是否包含另一个候选池，方程（S17）均适用。

MeasuredSelector结果。 主论文表3中观测到的Long得分比较了共享合法池上的排序规则。有限池保证假设源效用有界且采用条件抽样。证明并不要求代理指标按照映射在该任务上的期望效用对其进行排序。

表 S64. 八个工作负载上的更强代理选择器（%）。

工作负载	多输出重建	对角海森矩阵	Task
Long	91.79	92.11	93.67
Wall-L	92.48	92.74	94.36
Push-L	88.54	88.93	90.37
TwoRoom	84.15	84.62	86.18
Cube	82.13	82.49	84.01
Push-D	90.54	90.88	92.29
排序†	89.82	90.03	91.46
Windows†	35.86	36.10	37.59

任务得分为 96 个源区块的均值。

表 S65. 固定映射内核时间和临时存储空间。

工作负载	适配器时间	联合时间	适配器临时存储	联合临时存储
Push-L	30.16 ± 0.55	27.30 ± 0.61	0.344	0.247
Cube	47.47 ± 1.12	43.30 ± 1.03	0.703	0.329
Long	190.04 ± 4.88	171.51 ± 4.79	9.250	1.235
Wall-D	67.86 ± 1.56	61.12 ± 1.52	0.438	0.186
PointMaze	113.94 ± 2.89	103.08 ± 2.51	0.875	0.368

内核耗时为 64 次运行（μs）的均值和样本标准差。临时存储列以二进制兆字节为单位给出测量值。

J.5. FGR 中的分组恒等式、溢出与舍入

假设 2（量化算子条件）。 维度 $m, n_{\text{out}}, d_{\text{in}}$ 均为正整数，其中 $1 \le i \le m$ 和 $1 \le j \le n_{\text{out}}$。有限集合 $\mathcal{G}$ 个非空组对 $\{1, \dots, d_{\text{in}}\}$ 进行分区。整数编码、零点和正缩放因子定义了 $\widehat{a}_{ik} = s^a_{ig}(q^a_{ik} - z^a_{ig})$ 和 $\widehat{w}_{jk} = s^w_{jg}(q^w_{jk} - z^w_{jg})$，适用于 $k \in g$。参考实现和 FGR 使用相同的编码、缩放因子、零点、偏置、分组及最终输出数据类型。整数分组部分和不会溢出。生产者可以在本地保留部分累加器，并直接写入每个结果，而无需为完整输出分配转换缓冲区。

命题 1（FGR 保持量化线性算子）。 在假设 2下，在生产者内部应用每个组的缩放因子乘积，计算得到的实数运算线性算子与对重构的量化操作数进行乘法得到的结果相同。重新缩放和转换可以在不使用与输出同等大小的全局张量的情况下执行。

证明。 固定 i， j 并将点积按组展开。

$$y_{ij} = \sum_{k=1}^{d_{\text{in}}} \widehat{a}_{ik}\widehat{w}_{jk} + \beta_j \tag{S26}$$

$$= \sum_{g \in \mathcal{G}} \sum_{k \in g} s^a_{ig}(q^a_{ik} - z^a_{ig}) s^w_{jg}(q^w_{jk} - z^w_{jg}) + \beta_j \tag{S27}$$

$$= \sum_{g \in \mathcal{G}} s^a_{ig} s^w_{jg} \sum_{k \in g} \bar{q}^a_{ik}\bar{q}^w_{jk} + \beta_j. \tag{S28}$$

每个缩放因子在其组内保持不变，因此得到主论文中的公式（7）。构造性调度计算一个分组部分和 p_{ijg}，对其进行缩放，再将其加到局部输出累加器中，并将部分和存储空间用于下一个组。然后，它加上 β_j 并写入输出数据类型。按输出元素或受支持的输出块重复此调度，可避免使用与输出一样大的转换张量。我们仍会计入内核使用的寄存器、共享内存及任何其他空间。 □

分组边界与溢出。 对于任意部分和 p_1， p_2，恒等式 $s_1p_1 + s_2p_2 = s(p_1 + p_2)$ 对所有部分和都成立，当且仅当 $s_1 = s_2 = s$。因此，合并分组后再进行一次重新缩放通常是无效的。对于 $b_{\text{acc}} \ge 2$ 位的有符号整数累加器，定义 $A_g = \max_{i,\ k \in g} |\bar{q}^a_{ik}|$和 $W_g = \max_{j,\ k \in g} |\bar{q}^w_{jk}|$。若要保证每个乘积及其所有部分和都能表示且不发生溢出，则

$$|g| A_g W_g \le 2^{b_{\text{acc}}-1} - 1. \tag{S29}$$

公式 (S29) 可通过将任意前缀的大小界定为整个组中绝对乘积之和来得到。若不满足此条件，实现必须拆分求和，或使用足够宽的累加器，以容纳每个部分结果。

命题 2（缩放累加的前向误差界）。 固定一个输出 (i, j)，并记 $\nu = |\mathcal{G}|$ 和 $t_g = s^a_{ig}s^w_{jg}p_{ijg}$。假设部分和为精确整数，且转换到累加格式时精确。在两条路径上使用相同的表示缩放因子和偏置常数。每个 t_g 由两次舍入乘法计算，然后依次对 ν 项及偏置求和。每个运算都遵循单位舍入误差为 $0 < u_{\text{fp}} < 1$ 的相对误差模型，且不会发生溢出或下溢。对于 $(\nu+2)u_{\text{fp}} < 1$，定义 $\gamma_h = hu_{\text{fp}}/(1-hu_{\text{fp}})$、$h = 0$、.....、$\nu + 2$。对于计算所得的值 $\widetilde{y}_{ij}$，在将输出转换为最终类型之前，有如下界：

$$|\widetilde{y}_{ij} - y_{ij}| \le \gamma_{\nu+2} \left(\sum_{g \in \mathcal{G}} |t_g| + |\beta_j| \right). \tag{S30}$$

如果最终数据类型转换遵循相同的舍入模型 $0 < u_{\text{out}} < 1$，则其结果 y^{out}_{ij} 满足

$$|y^{\text{out}}_{ij} - y_{ij}| \le (1 + u_{\text{out}})|\widetilde{y}_{ij} - y_{ij}| + u_{\text{out}}|y_{ij}|. \tag{S31}$$

证明。 h 舍入因子 $(1 + \epsilon_r)$ 的乘积 $|\epsilon_r| \le u_{\text{fp}}$与 1 的差异至多为 γ_h， 当 $hu_{\text{fp}} < 1$时成立。对于上偏差，展开 $(1+u_{\text{fp}})^{\ h}$，并利用 $\binom{h}{k} \le h^k$ 将其界定为 $(1-hu_{\text{fp}})^{-1}$。下偏差由 $(1-u_{\text{fp}})^{\ h} \ge 1 - hu_{\text{fp}}$得出。每个组项涉及两次乘法和至多 ν 次加法。偏置至多参与 ν 次加法。若按计算顺序展开求和，则每个精确项的相对误差有界于 $\gamma_{\nu+2}$。由三角不等式可得方程（S30）。对于类型转换，我们加上最终舍入步骤带来的误差 $y^{\text{out}}_{ij} = \widetilde{y}_{ij}(1 + \epsilon_{\text{out}})$， 而$|\widetilde{y}_{ij}| \le |y_{ij}| + |\widetilde{y}_{ij} - y_{ij}|$ 则给出了相对于精确量化结果的输出类型转换误差上界。 □

表 S66。使用调优后的同映射适配器进行完整运行比较。

工作负载	适配器 s	联合 s	适配器 MB	联合 MB	时间 ↓ (%)	峰值 ↓(%)
Push-L	3.640	3.3420	175.40	166.40	8.19	5.13
Cube	4.680	4.2080	176.50	164.00	10.09	7.08
Long	7.530	6.5950	7815.00	7111.80	12.42	9.00
Wall-D	5.480	5.0520	928.50	873.80	7.81	5.89
PointMaze	6.790	6.0060	1009.00	922.20	11.55	8.60

每个工作负载的均值涵盖 64 次配对运行。所有路径均未改变任务结果。为与适配器进行比较，E 记录每对路径的时间和峰值内存。

表 S67. 分布偏移下的任务质量（%）

工作负载	原始冻结	分布外 FP32	分布外 W4	分布外冻结	分布外重新选择
Long	93.90	87.74	82.86	88.21	92.81
TwoRoom	86.27	78.39	74.16	79.57	84.56
Windows	37.70	29.68	26.47	30.55	36.45

表 S68。跨任务迁移与目标重新选择（%）。

源 → 目标	迁移	目标重新选择
目标 → Long	92.13	93.56
Wall-L → Push-L	87.82	90.14

每个目标有 48 个区块，每个区块包含 512 个最终源簇。

该界限适用于所述算术条件和求和顺序。非精确的整数转换需要额外的转换误差项；而融合操作或改变操作顺序的内核，则需要单独计算运算计数或进行数值检查。实数运算等价是指两条路径使用相同的量化操作数。与原始 FP32 运算符的一致性不属于此恒等式的范畴；此外，量化路径之间若要逐位一致，还要求算术顺序和舍入方式完全相同。附录G.6 报告了对受测路径进行实验所得的容差。整数乘积的恒等式不包括以不同精度运行的受保护操作。

J.6. DHR 中的逻辑索引与消费者等价性

假设 3（对上下文分段消费者的条件）。 $J \geq 1$ 分段的长度 ℓ_j均为正整数，数据类型兼容，且非上下文维度相匹配。偏移量为 $o_0 = 0, o_j = \sum_{r=1}^{j} \ell_r$。在最终读取完成前，分段长度、偏移量和内容均保持有效且不可变。消费者读取张量值和逻辑元数据，包括位置、掩码和量化缩放因子。它既不检验存储空间是否连续，也不跨分段执行指针运算。所有读取均不会更改存储的输入值。

命题3 (DHR 保持兼容的消费者)。在假设3下，虚拟拼接和物化拼接在每个逻辑索引处都返回相同的值和元数据。因此，仅通过此接口读取的确定性消费者对于这两种布局，都会返回相同的数学结果。

证明。 长度为正意味着 $0 = o_0 < o_1 < \cdots < o_J$。每个有效的逻辑上下文索引 $p \in \{0, \ldots, o_J - 1\}$ 都恰好属于一个区间 $[o_{j-1}, o_j)$ 。因此，在非上下文坐标固定时，逻辑位置 p处的值为

$$\mathbf{X}[\ldots, p, \ldots] = \mathbf{X}_j[\ldots, p - o_{j-1}, \ldots]. \tag{S32}$$

DHR 通过返回公式（S8）右侧的结果以及相同的逻辑元数据来实现公式（S32）。从相同的消费者状态出发，对接口读取和确定性状态转移进行归纳，可得数学输出相同。假设的接口不允许观测或修改存储地址，否则会使这一归纳失效。当浮点运算顺序发生变化时，我们会通过单独的测试检查两个输出的一致程度。 □

注意力的全局归一化。 设查询 $\mathbf{q}_{\text{att}} \in \mathbb{R}^{d_k}$ 和逻辑键 $\mathbf{k}_p \in \mathbb{R}^{d_k}$ 的键维度为正整数 d_k，值 $\mathbf{v}_p \in \mathbb{R}^{d_v}$ 的维度为正整数 d_v。令 $\mu_p \in \mathbb{R} \cup \{-\infty\}$ 编码位置和掩码贡献，并设 $a_p = \mathbf{q}_{\text{att}}^{\top} \mathbf{k}_p / \sqrt{d_k} + \mu_p$。假设所有未掩码的 logit 均为有限值，且至少有一个索引未被掩码。在这一单一全局归一化下，其输出为

$$\frac{\sum_p e^{a_p - a_{\max}} \mathbf{v}_p}{\sum_p e^{a_p - a_{\max}}}, \qquad a_{\max} = \max_p a_p. \tag{S33}$$

每个求和都覆盖完整的逻辑上下文。拼接各分段归一化结果会改变算子。对于每个未被完全屏蔽的分段 r，定义 $m_r = \max_{p \in r} a_p$、$l_r = \sum_{p \in r} e^{a_p - m_r}$ 和 $\mathbf{u}_r = \sum_{p \in r} e^{a_p - m_r} \mathbf{v}_p$。对于两个非空聚合量，我们按以下规则合并它们的和，

$$m = \max(m_1, m_2), \tag{S34}$$
$$l = e^{m_1 - m} l_1 + e^{m_2 - m} l_2, \tag{S35}$$
$$\mathbf{u} = e^{m_1 - m} \mathbf{u}_1 + e^{m_2 - m} \mathbf{u}_2. \tag{S36}$$

表 S69. 两种资源预算下的短上下文 Wall-D。

路径	第95百分位（秒）	峰值（MiB）	5.10s/985MiB	5.50s/840MiB
打包 (q_{00})	4.95	972.4	Yes	No
仅 FGR (q_{10})	5.04	913.7	Yes	No
仅 DHR (q_{01})	5.21	864.2	No	No
FGR 和 DHR (q_{11})	5.33	821.6	No	Yes

表 S70. 对完整 Long 模拟器批次进行计时（秒）。

路径	模型	环境	规划/调度	I/O	热启动总计	冷启动总计	成功
BF16	774.450	108.700	52.300	38.800	974.250	1022.850	476/500（95.2%）
PTQ4WAM	476.400	108.700	52.300	38.800	676.200	7985.700	480/500（96.0%）

表 S71. Long 模拟器热启动与冷启动批次（秒）。

路径	热启动均值（秒）	预热标准差（秒）	准备（秒）	冷启动（秒）	成功数 / 6000
BF16	973.047167	10.268617	48.6	1021.647167	5706（95.1000%）
PTQ4WAM	675.730750	7.162797	3469.4	4145.130750	5719（95.3167%）

表 S72. 12 个模拟器批次的阶段时间和成功次数。

批次	BF16 M	BF16 E	BF16 P	BF16 输入/输出	W M	W E	W P	W 输入/输出	BF16 存活集	W S
01	777.759	108.109	53.463	38.310	479.952	107.690	53.908	38.996	476	479
02	779.305	107.658	52.370	39.406	479.682	107.849	52.983	39.022	473	474
03	767.873	107.517	52.799	37.335	468.629	108.379	52.600	36.877	480	478
04	772.542	106.777	50.450	38.771	475.733	105.368	49.995	38.970	469	475
05	771.305	109.101	50.705	37.653	470.401	109.755	51.184	37.910	478	480
06	790.122	107.024	52.910	37.599	487.198	106.069	52.112	38.048	475	475
07	780.937	110.874	51.689	37.584	481.305	110.648	50.710	38.427	482	483
08	763.983	107.035	54.398	41.175	465.182	107.858	53.945	40.677	474	471
09	771.860	107.719	52.190	39.427	482.064	108.430	51.764	39.200	471	475
10	785.956	104.159	52.766	39.075	487.634	103.653	52.730	38.900	479	478
11	783.448	103.997	52.897	39.269	483.046	104.242	53.791	39.718	477	477
12	751.244	109.480	51.600	36.941	468.527	109.379	50.947	36.682	472	474

每个批次包含 500 个新回合。M、E 和 P 分别报告模型执行、环境推进和规划/调度所用的时间。S 统计机器人达成目标的回合数。

通过代换，可将 l，$\mathbf{u}$ 表示为相对于 m 的完整并集的指数分母和分子。对分段应用归纳法可得全局表达式 $\mathbf{u}/l$，其中 $l > 0$。完全屏蔽的分段贡献为零，可以跳过。这个等式明确了分段注意力使用方所需的全局归一化。

存储生命周期。 保留原始分段可能会延长其生命周期，使其存活时间超过连续复制释放它们的时点。额外的存活存储空间、地址元数据以及任何额外工作区都会计入 H_t 存活集模型。同一底层存储的别名只计一次。索引恒等式保证逻辑等价。峰值判据决定这些分配变化是否会降低全局内存使用量。

J.7. 完整存活集证明与扩展

假设4（完整对齐分配核算）。 非空有限事件集 $\mathcal{T}$ 包含了与任一路径相关的每个恒定分配区间的代表事件。逻辑事件使调度对齐，但不要求墙上时钟时间戳相同。基线字节数和变换后字节数 M_t，M_t' 均在相同的检测机制下统计存活设备分配，包括持久存储和工作区。我们统计所有变化，包括每个分段的存活时长以及调度分配了哪些缓冲区。每项变化都计入移除的存储空间 $D_t \geq 0$ 或新增的存储空间 $H_t \geq 0$，因此 $M_t' = M_t - D_t + H_t \geq 0$。如果多个张量别名到同一分配，则存活字节数中只计一次。

全局峰值判据。 避免中间结果并不能确保工作负载峰值更低，因为其他阶段可能占据主导并内存使用量更高。因此，我们跟踪每个事件同时分配的存储空间，即活动

表 S73。选择与执行证明中使用的符号。

符号	定义
$w, \mathcal{U}_w, N_w$	工作负载、符合条件的单元和符合条件的单元数量。
$c, b_i, \mathcal{C}_w, K$	精度映射、单元位宽、有序合法池和池基数。
$A_w(c)$ $S_w(c)$ B_w	冻结的打包产物、其序列化字节和正目标预算。
ε_B, $\varepsilon_{\mathrm{rec}}$	预算容差（10^{-3}）和重构稳定项（10^{-8}）。
Θ_w, Θ_w, Γ_w	预设提案区域、收缩区域和固定合法池生成器。
$I_w, \phi, \xi, \mathcal{A}_\phi$	开发前元数据、智能体配置、提案随机性和智能体映射。
$\mathscr{F}_w, \mathcal{P}_w, Z_j, n$	开发前信息σ-域、源分布、源簇和独立的开发簇数量。
$v(c, Z)$ $q(c)$ $\widehat{q}_n(c)$	有界源效用、条件总体效用和经验均值。
c_R, c_H, c_T, $c^\star$	重构、海森矩阵、经验任务和总体最优选择。
δ, η_K	（0，1）内的失败概率和一致集中半径。
$\mathcal{C}_w^B, \mathcal{C}_w^A$	Broad参考值和代理生成的合法池。
$q_B^\star, q_A^\star, \rho_A$	相对于 Broad 的池最优效用和正部覆盖损失。
$\mathbf{W}_i$, $\widehat{\text{宽}}_i$, 高$_i$, d_i	原始/重构权重、校准海森近似以及权重向量维度。
$m, n_{\mathrm{out}}, d_{\mathrm{in}}$	输入行、输出通道和归约宽度。
$\mathcal{G}, g, p_{ijg}, \nu$	组划分、一个组、整数群部分和组数。
$q^a, q^w, z^a, z^w, s^a, s^w, \beta$	激活/权重码、零点、正尺度和偏置。
$J, \ell_j, o_j, d_{\mathrm{ctx}}$	分段数量、长度、前缀偏移量和上下文轴。
$\mathcal{T}, M_t, D_t, H_t$	对齐事件、基线活动字节数、移除字节数以及新增的工作空间或延长生命周期字节数。
d_t, g_t, P, P'	净节省量、基线余量以及基线/转换后的全局峰值。
F,P, Q T_r, M_r	FP32、模块关闭对照、最终路径、路径时间和路径峰值分配。

集。设$\mathcal{T}$为覆盖两条路径中所有峰值相关活动集合的非空有限对齐逻辑分配事件集合。在事件 t处，记基线活动字节数为 $M_t \geq 0$，移除字节数为 $D_t \geq 0$，记工作区或延长的生命周期所增加的字节数为 $H_t \geq 0$。将共享存储空间仅计数一次，并核算每项变化，得到 $M_t' = M_t - D_t + H_t \geq 0$。定义净节省量 $d_t = D_t - H_t$、全局基线峰值$P = \max_t M_t$以及每事件余量 $g_t = P - M_t$。

定理2（完整活动集合核算恒等式）。在完整的对齐核算下（假设 4），变换后的峰值 $P' = \max_t M_t'$满足

$$P - P' = \min_{t \in \mathcal{T}} (g_t + d_t). \tag{S37}$$

因此，仅当 $d_t > -g_t$ 在每个事件处成立时，$P' < P$才成立。每个原始峰值都必须缩减。在其他事件处，净增量必须严格小于原始余量。

当公式（S37）中的核算仅在一个阶段发生变化时，主论文中的公式（9）会根据竞争性未变峰值表示峰值减少。以下推论刻画了局部节省何时会完整体现于模型峰值中。

定理2的完整证明。令 $P = \max_t M_t$，$g_t = P - M_t$ 和 $d_t = D_t - H_t$，我们得到

$$P' = \max_t (M_t - D_t + H_t) \tag{S38}$$

$$= \max_t \{P - (g_t + d_t)\} \tag{S39}$$

$$= P - \min_t (g_t + d_t). \tag{S40}$$

由于 $\mathcal{T}$是有限且非空的，因此两个极值都能取到。因此，$P - P' = \min_t (g_t + d_t)$，当且仅当对每个事件均有 $d_t > -g_t$ 时，该值为正。在基线最大值处，$g_t = 0$ 和 d_t 必须为正。在非峰值事件处，只有当负净节省的绝对值小于可用余量时，才允许出现负净节省。若从 $\mathcal{T}$ 中遗漏仅由变换产生的峰值，则前提将不成立。 □

推论 2（单阶段饱和）。设某阶段在变更前占用$A \geq 0$ 字节，并节省 d_{save} 字节，其中$0 \leq d_{\mathrm{save}} \leq A$。设 $C \geq 0$ 为最大未变分配；若没有剩余项，则为 $C = 0$。那么

$$P = \max(A, C), \quad P' = \max(A - d_{\mathrm{save}}, C),$$
$$P - P' = \begin{cases} \min(d_{\mathrm{save}}, A - C), & A \geq C, \\ 0, & A < C. \end{cases} \tag{S41}$$

证明。当 $A < C$时，C 支配两条路径。当$A \geq C$且 $A - d_{\mathrm{save}} \geq C$时，缩减量为 d_{save}。否则，新的峰值为 C，缩减量为 $A - C$。实际节省量取局部缩减量与竞争阶段设定峰值前的余量两者中的较小值。 □

表中的 Wall-D 和 PointMaze 记录S55 显示，Wall-D 节省了 151.3 MiB，Point-Maze 节省了 169.3 MiB。两种情况下，新的峰值均由一个未变的阶段设定。要完成核算，必须计入未变阶段中的分配量，因为其中一个阶段可能成为新的峰值。

J.8. 系统与模块分解

分离总体效应与增量效应。 为区分整体部署增益与执行变更带来的额外效应，我们固定 $A_w(c_T)$，并比较其兼容的执行路径。令 F 表示 FP32，P 表示打包模块关闭对照，Q 表示最终启用 FGR+DHR 的路径。对于路径 r，令 $T_r > 0$ 表示其测量时间，$M_r > 0$ 表示其分配的峰值字节数。这些测量值给出了整体速度和内存比率，

$$S_{\text{total}} = T_{\mathsf{F}}/T_{\mathsf{Q}}, \qquad R_{\text{total}} = 1 - M_{\mathsf{Q}}/M_{\mathsf{F}}. \tag{S42}$$

公式（S42）中的比率是在初始化后，对相同固定工作负载的 GPU 重放进行比较，反映完整路径相对于 FP32 的表现。它们不包括离线构建和评分。通过在相同制品上比较P与 Q （公式（S43）），可测量模块带来的额外效应。FP32 与对照组之间的差异还包含启用模块前所做的变更，因此，仅靠权重压缩无法解释这两条路径之间的全部差异。

对于冻结的制品，P 仅禁用受测模块，Q 则启用该模块。启用该模块所导致的已分配内存和相对时间变化为

$$R_{\text{module}} = 1 - M_{\mathsf{Q}}/M_{\mathsf{P}}, \qquad \Delta T_{\text{module}} = (T_{\mathsf{Q}} - T_{\mathsf{P}})/T_{\mathsf{P}}. \tag{S43}$$

对于一组三者对齐的 FP32/对照组/FINAL 测量值，定义

$$S_{\text{pre}} = T_{\mathsf{F}}/T_{\mathsf{P}}, \qquad S_{\text{module}} = T_{\mathsf{P}}/T_{\mathsf{Q}}, \qquad R_{\text{pre}} = 1 - M_{\mathsf{P}}/M_{\mathsf{F}}. \tag{S44}$$

所有时间和峰值均为正。通过约去两项比较共有的端点，这些比率可以组合：

$$S_{\text{total}} = S_{\text{pre}} S_{\text{module}}, \qquad 1 - R_{\text{total}} = (1 - R_{\text{pre}})(1 - R_{\text{module}}). \tag{S45}$$

此外，$S_{\text{module}} = 1/(1 + \Delta T_{\text{module}})$。正的计时开销会使模块因子小于 1，即使完整路径比 FP32 更快也是如此。模块启用前的项包含 FP32 与打包对照组之间的所有差异，因此无法单独分离出权重压缩的影响。应用方程（S45）需要一个共享的中间端点。如果分别取中位数，或更改设备或重放输入，这一要求可能无法满足。

K. 额外重复测量与统计摘要

测量单位和参考端点。 主论文的表 1 报告了固定工作负载下 GPU 重放 FP32 对比结果，并单独测量了任务质量。表 S74汇总了八种工作负载的准备成本，以及五种工作负载中 64 个配对模块-对照区块的准备成本。每个比率均为对照组除以对应的参考值。对于每项指标，我们报告跨工作负载的中位数，随后在方括号中给出最小值和最大值。表 S75 具体说明了四项配对检验，表 S76则给出了用于计算区块差值的原始配对值。

表 S74. 按协议进行的准备和算子比较。

比较 / 指标	设置对照/参考比
准备 / Broad-48 至 Agent-12	8 3.888 [3.325, 3.982]
模块 / 时间	5 1.253 [1.215，1.271]
模块/峰值分配	5 1.346 [1.313，1.380]

任务质量与重复提案的检验。原生设置下的质量根据 48 个区块的原生测量结果汇总，每个区块包含 512 个源簇；选择和候选池结果则使用 96 个区块，每个区块包含 128 个开发源簇和 256 个最终源簇。重复执行、比较和窗口仍按源分组。以下配对推断采用四项预定义比较，每项包含 12 个配对区块。双侧精确符号检验和 Holm 校正仅适用于这四项比较构成的检验族。单独的 S/N/E 协议针对每项任务采用 12 个配对区块，Holm 检验族大小分别为 8、8、5（时间）和 5（分配峰值）。额外的 A/F/R 协议采用 12 个配对区块，比较数量分别为 3、1 和 8。R 将时间和内存对比合并到一个检验族中。平均配对差采用 95% Student-t 区间（$df = 11$），分析单位为区块内差值。

表 S75。四项配对检验的协议。

字段	记录值
比较族大小	4
每项比较的配对区块数	12
每项比较的胜出数/失败数/平局数	12/0/0
双侧精确符号检验 p	0.00048828125
经 Holm 校正的 p	0.001953125
显著性水平 α	0.05

95% 区间使用 $d \pm 2.\overline{2009851601} s_d/\sqrt{12}$，自由度为 11。对于 Long 时间对比，均值比给出的加速比为 $1.270963\times$，时间缩减为 21.319494%；而所有配对的平均加速比和平均缩减分别为 $1.271686\times$ 和 21.336099%。

全部四项配对比较均为12次胜出、0次失利、0次并列。双侧精确符号检验得出 p =0.00048828125，四个案例的结果均相同。Holm校正后的对应数值为 $p = 0.001953125$。表S76 给出了原始观测值。表S77 列出了 Long 上智能体和固定候选名单的逐区块结果。

表 S76。四项预先指定的测试的配对观测。

区块	A 对照	A 任务	B 对照	B 任务	C q_{00}	C q_{11}	D FP32	D PTQ4WAM
1	701	716	640	654	8.560	6.704	200	243
2	705	718	614	623	8.119	6.451	197	246
3	704	720	632	647	8.464	6.607	193	242
4	708	721	631	647	8.242	6.665	190	237
5	704	719	637	644	8.599	6.917	208	244
6	709	716	641	649	8.170	6.205	195	240
7	718	732	628	641	8.426	6.647	199	241
8	712	721	642	657	8.672	6.981	213	244
9	701	711	642	654	8.103	6.200	201	240
10	713	722	642	656	8.550	6.785	209	243
11	711	726	623	630	8.404	6.543	208	242
12	703	711	630	640	8.275	6.435	184	233

A 和 B 比较 Long 和 Cube 上的对角与任务（总计 768 个）。C 比较 Long 的重放时间（秒）。D 比较 FP32 和 PTQ4WAM 在 Wall 上的成功次数，共 256 个状态。

表 S77。Long 上 Agent-12 与 Fixed-12 的比较（总计 768 个）。

区块	固定	智能体	智能体–固定（pp）
1	718	726	1.042
2	722	727	0.651
3	719	726	0.911
4	716	722	0.781
5	724	733	1.172
6	715	719	0.521
7	721	728	0.911
8	717	722	0.651
9	723	731	1.042
10	720	726	0.781
11	719	723	0.521
12	721	727	0.781

配对比较中 Agent–Fixed 增益的平均值为 0.814 百分点（样本标准差 0.209 百分点，95% Student-t CI $[0.681, 0.946]$ 百分点，12 次胜出、0 次失利、0 次并列，名义双侧精确符号检验 $p = 0.00048828125$）。这项额外审计仅作描述，不属于四项比较的 Holm 检验族。

表 S78 统计描述性比较的数量。表 S79 展示了改变选择方式或提案规则时任务得分的变化。

表 S78。各项研究中的比较次数。

比较	计数
选择方式与候选池	56
执行控制	120
FGR/DHR 模块控制	10
偏移与重新选择	3
Wall 干预	6
总计描述性效应	195

这 195 个描述性效应使用 96 个区块进行选择，使用 64 次配对运行进行执行和模块比较，并使用 48 个区块进行分布偏移和 Wall 比较。未经 Holm 校正的 p 值或置信区间被分配给这 195 个描述性摘要。S/N 和 A/F/R 协议以及测得的 E 记录各自拥有独立的配对推断检验族。`-J` 质量和 `-X` 计时检验分别在表S6 和S19中报告。具有原始配对观测值的四项检验构成一个单独的预先指定的检验族。统计显著性仅针对已确定的配对检验族进行判定。每项检验仅适用于其所属的检验族。我们分别报告描述性效应以及跨工作负载的中位数。

绝对执行端点。对于相同的固定工作负载，表格 S80 和 S81 给出了平均重放时间和峰值分配。表格 S82 报告了根据这些绝对端点计算出的 120 项缩减。所有对应比率均精确到小数点后两位。这些记录采用固定输入的执行协议。原生质量队列和配对排名实验使用了不同的观测数据。

表S83 比较各执行模块与打包路径。表 S84衡量分布偏移下的传输和重新选择。表 S85给出了规划器在 Wall 上匹配干预的结果。

表S86 总结了四项预先指定的配对检验。对于独立部署队列，表 S87和S88 展示了边际计数能够支持哪些符号检验结果。表S89 展示了随着开发样本增多，界限下降了多少。

对于 $n_{\mathrm{dev}} = 128$，$K = 12$ 个不同的评分对对应的理论界为 31.059 个百分点，$K = 48$ 个不同的评分对对应的理论界为 34.370 个百分点。如果源之间条件独立，在所述失败概率下，$n_{\mathrm{dev}} \geq 123476$ 时的 $K = 12$ 界最多降至 1 个百分点。确定候选池大小时，任务质量相同的路径只计一次。路由特定质量不同时，每个映射最多计入四条路径。该理论界适用于我们评分的映射。

表 S79。八种工作负载下的选择效应和提议效应。

	任务减去选择器			智能体减去提议规则			
工作负载	海森矩阵	重建	对角	固定	随机	广泛	敏感性
Long	+3.10	+3.83	+1.56	+0.81	+1.27	-0.07	+0.59
Wall-L	+3.01	+3.81	+1.62	+0.65	+1.10	-0.05	+0.49
Push-L	+2.85	+3.58	+1.44	+0.95	+1.38	-0.08	+0.70
TwoRoom	+2.63	+3.57	+1.56	+0.73	+1.22	-0.04	+0.51
Cube	+2.65	+3.54	+1.52	+0.92	+1.36	-0.08	+0.60
Push-D	+2.74	+3.63	+1.41	+0.65	+1.06	-0.06	+0.48
排序	+3.14	+3.83	+1.43	+0.72	+1.09	-0.04	+0.53
Windows	+2.80	+3.50	+1.49	+0.89	+1.55	-0.09	+0.56

每个单元格是一个 在 96 字节锁上的描述性配对差值（百分点） 锁。

表 S80. 各工作负载的平均固定工作重放时间（秒）。

Task	BF16-opt	W4-native	W8-native	Packed-same-map	PTQ4WAM
空间	47.114365	40.091384	44.239130	37.739917	33.781000
物体	60.436709	48.881495	48.429060	43.567925	38.196000
Goal	49.746009	45.397351	43.645740	37.377317	34.275000
Long	9.860945	8.564935	8.640115	8.382000	6.595000
Wall-L	19.292324	12.732934	14.816054	11.663010	10.632000
Push-L	4.918322	4.245966	4.673472	4.140000	3.342000
TwoRoom	17.299841	14.264861	13.777467	11.897554	10.847000
Cube	7.182113	5.175255	5.270541	5.112000	4.208000
Reacher	5.613182	3.766312	4.401921	3.452592	3.117000
Wall-D	8.211964	6.142996	6.805874	6.330000	5.052000
PointMaze	10.734584	7.888101	7.970803	7.560000	6.006000
Push-D	29.315766	24.711007	24.729215	21.776792	20.137000
Meta-World Reach	87.375092	75.252602	74.929221	66.980002	59.284000
排序	22.211990	14.733501	15.313084	13.654124	12.301000
Windows	83.850797	67.366327	75.813042	58.944620	54.176000

L. 序列化部署注册表和配对源记录

L.1. 合格单元表示与制品计数

我们部署了原始的预训练 Fast-WAM、LeWM、DINO-WM 和 V-JEPA 2-AC 检查点，并保留了其模型架构、输入处理，以及每个检查点所提供的规划器或任务接口。量化作用于已加载的参数张量。零起始的单元 ID `.u000`、`.u001`等用于索引包含实际状态_字典键、形状、数据类型、参数切片、存储别名和捕获的执行图节点的注册表条目。张量形状和图连通性取自模型。表S90 列出了各系列的注册表命名空间。表S91则标明了受保护参数和精度分配。序列化候选包占用

$$S = S_{\mathrm{W4}} + S_{\mathrm{W8}} + S_{\mathrm{scale}} + S_{\mathrm{zero}} + S_{\mathrm{protected}} + S_{\mathrm{header}} + S_{\mathrm{pad}}. \tag{S46}$$

在公式（S46）中， P_b 是以位宽 b存储的参数数量。组件字节数包括组尾部，打包尾部、受保护参数、头部和对齐填充。别名存储只计数一次。记录账本使用 FP16 缩放因子、 $G = 32$、零零点字节数和 $P_{32} = 0$。预算在候选搜索之前固定，并且相对容差 10^{-3} 适用于每个候选项。表 S92 中选定的部署制品达到 $S = B$。存储字节数使用 MB $= 10^6$ 字节和 MiB $= 2^{20}$字节。W4 模块比例为 N_4/N。全参数比例为 $P_4/$（$P_4 + P_8 + P_{16} + P_{32}$）。W4 载荷比例为（$P_4/2$）/（$P_4/2 + P_8 + 2P_{16} + 4P_{32}$）。压缩使用 4（$P_4 + P_8 + P_{16} + P_{32}$）$/S$（FP32 权重载荷，而非特定 FP32 文件的大小）。每个合格参数的字节数为 $S/$（$P_4 + P_8$）。原生激活值为 A16。共享池研究使用 A8，但受保护的 Windows 算子除外，它们以 A16 运行。我们使用 upstream_ref 定位上游代码，并使用修订版本固定版本。我们使用 checkpoint_sha256 验证检查点，并使用 co nfig_sha256 验证配置。此注册表身份使用 registry_ref 和 registry_sha256。单元条目使用 unit_id 、 state_dict_ key、形状、数据类型、切片、别名和 graph_nodes，将每项精度分配同时绑定到存储其所持有

表 S81. 各工作负载的峰值分配内存均值（MB）。

Task	BF16-opt	W4-native	W8-native	Packed-same-map	PTQ4WAM
空间	10811.269975	7746.534892	8934.250765	7299.602499	6427.300000
物体	10556.214818	8870.621324	10117.526976	7866.001587	6938.600000
Goal	11028.455285	8105.564546	8939.044481	7330.781356	6511.200000
Long	12627.485795	8430.298720	10365.544381	9812.700000	7111.800000
Wall-L	299.305908	208.000000	246.239554	195.943699	176.800000
Push-L	258.064516	202.260848	216.553878	223.400000	166.400000
TwoRoom	312.176859	224.717659	227.196056	190.326087	175.100000
Cube	290.162774	204.718512	238.268197	215.400000	164.000000
Reacher	223.158217	172.773627	188.877366	168.458781	145.700000
Wall-D	1525.755195	1049.357512	1294.710327	1190.400000	873.800000
PointMaze	1564.376590	1203.759300	1326.524741	1241.600000	922.200000
Push-D	1584.101962	1124.389954	1229.147690	1071.339458	944.600000
Meta-World Reach	1470.055600	1197.463768	1233.373317	1047.543582	925.400000
排序	2564.234668	1825.269135	2255.042385	1746.745160	1542.900000
Windows	3569.309300	2878.682619	3195.658383	2429.269352	2237.600000

表 S82. 固定工作负载下的时间和内存分配减少（%）。

	重放时间				峰值分配			
工作负载	BF16-opt	W4	W8	打包	BF16-opt	W4	W8	打包
空间	28.30	15.74	23.64	10.49	40.55	17.03	28.06	11.95
物体	36.80	21.86	21.13	12.33	34.27	21.78	31.42	11.79
Goal	31.10	24.50	21.47	8.30	40.96	19.67	27.16	11.18
Long	33.12	23.00	23.67	21.32	43.68	15.64	31.39	27.52
Wall-L	44.89	16.50	28.24	8.84	40.93	15.00	28.20	9.77
Push-L	32.05	21.29	28.49	19.28	35.52	17.73	23.16	25.51
TwoRoom	37.30	23.96	21.27	8.83	43.91	22.08	22.93	8.00
Cube	41.41	18.69	20.16	17.68	43.48	19.89	31.17	23.86
Reacher	44.47	17.24	29.19	9.72	34.71	15.67	22.86	13.51
Wall-D	38.48	17.76	25.77	20.19	42.73	16.73	32.51	26.60
PointMaze	44.05	23.86	24.65	20.56	41.05	23.39	30.48	25.72
Push-D	31.31	18.51	18.57	7.53	40.37	15.99	23.15	11.83
Meta-World Reach	32.15	21.22	20.88	11.49	37.05	22.72	24.97	11.66
排序	44.62	16.51	19.67	9.91	39.83	15.47	31.58	11.67
Windows	35.39	19.58	28.54	8.09	37.31	22.27	29.98	7.89

百分比对比 PTQ4 WAM 与各列基线的对比采用 64 次配对运行的均值比率。报告了 120 个描述性效应 ed
未对每项对比 分别进行检验。

参数的存储空间以及执行这些参数的算子 m.

表格S92 列出了序列化字节数。表格S93比较了文件大小、参数数量，以及各精度下权重字节所占的比例。

表 S94 将每个已注册制品与主论文表 1 中测得的部署结果关联起来。对于结果 `-T1:{task}`，质量和回放运行标识符分别为`-T1:{task}:Q`和 `-T1:{task}:R`。候选标识符仅在各自的协议、任务、区块和候选池中有效。键 运行_ID 用于标识一次运行。我们通过检查点哈希值标识模型 检查点_`sha256` ，并通过 注册表_`sha256` 标识单元注册表。随后，我们将协议记录在协议 中，并将匹配的源区块记录在源_区块中。为重建选择结果，我们记录候选池的未-排序候选池_sha256、通过 selected_id 记录的所选映射，以及通过 bitset sha256 记录的位集。保存的模型_具有摘要 artifact_sha256，其配置具有摘要 config_sha256。我们使用 source_manifest_sha256 来恢复源，并使用 log_ref 来定位日志。最后，table_cell 用于标识表格标签、行和指标。部署键将候选池与候选标识符连接起来。注册表顺序和位集摘要共同将制品关联到执行它的运行。

表S95 展示了哪些字段将注册模型与其运行的源和结果关联起来。

表 S83. 打包路径的相对降幅（%）。

比较	n	减少率（%）
Push-L/QEF-与关闭状态的比较/时间_秒	64	7.41
Push-L/QEF-与关闭状态的比较/峰值_字节	64	17.10
Cube/QEF-与关闭状态的比较/时间_秒	64	9.31
Cube/QEF-与关闭相比/峰值_字节	64	16.22
Long/QEF-与关闭相比/时间_秒	64	10.53
Long/QEF-与关闭相比/峰值_字节	64	18.58
Wall-D/VCF-与关闭相比/时间 _S	64	7.62
Wall-D/VCF-与关闭相比/峰值_字节	64	13.39
PointMaze/VCF-与关闭相比/时间 _S	64	8.83
PointMaze/VCF-vs-off/峰值_字节	64	14.13

正值表示相对于打包映射在 64 次配对运行中的缩减幅度。存档记录将 FGR 标识为 QEF，并将 DHR 标识为 VCF。

L.2. 映射枚举、哈希值与标识符

注册种子为 20261115。每项任务都有一个独立的部署池，其本地地图标签为 C00。表 S39 中列出了所选记录 : S:Long:01:C20 和 :S:Wall-L:01:C34。只有当注册表顺序和有效位集摘要相匹配时，部署别名才表示同一地图。对于 Long 和 Wall-L 以外的任务，将符合条件的单元划分为参数计数类别 d^2 和 $4d^2$。在每个类别中，按照 ASCII 字符串 `|task|20261115|i` 的 SHA-256 值升序排列从零开始的索引；若哈希平局，则按索引排序，并将 W4 分配给表 S91 中最前面的$N_{4,\text{small}}$ 和 $N_{4,\text{large}}$ 个单元。计数满足$N_{4,\text{large}} = (P_4/d^2 - N_4)/3$ 和 $N_{4,\text{small}} = N_4 - N_{4,\text{large}}$，同时保留 N_4 和 P_4。保留 Long 和 Wall-L 的位集。这定义了候选池基准 C00。在单交换候选池中，枚举一个 W4 索引 i 与一个 W8 索引 j 的交换，且其实际参数数量为 $n_i = n_j$。按 SHA-256 摘要 `|task|20261115|swap|i|j` 排列交换，然后按 (i, j) 排序，并将最前面的 47 个不同交换命名为 C01–C47。每个交换都基于其冻结的 C00 进行。 $K = 1$ 任务仅包含 C00。每张地图的序列化字节数均与 C00 相同。9 个各含 48 张地图的候选池和 6 个单例候选池共计 438 张地图，分布在 15 个注册表中，且在各自候选池内均不重复（重复计数为 0，两两汉明距离至少为 2，最大相对字节预算误差为 0）。候选搜索上限为 100,000，最多允许两次修复尝试。无效标识符、重复地图或违反字节预算的情况均不予接受。: P 中使用的多半径池构造在附录 B.6 中另行说明。

位集按最高有效位在前的顺序书写，在规范单元顺序中，1 表示 W4，0 表示 W8。前N位为有效位，末尾填充位为零；SHA-256 是对有效 ASCII 位串计算得出，不含换行符，结果使用小写十六进制表示。-R 注册表包含 438 个映射。其完整标识符排列映射至 -A，其中 Long 的 C00 与 C20 互换，Wall-L 的 C00 与 C34 互换。其他标识符均保持不变。

因此，选定的 C20 和 C34 保留了表 S39 中 Long 和 Wall-L 的位集。我们为注册表和提案、`:A` 输入干预，以及用于测量协议中 Ranking 执行情况的映射使用不同的前缀 `:R`。

L.3. 智能体输入/输出契约与源清单

智能体接收校准注册表、符合条件的单元形状、架构连接、量化约束、确切预算以及 48 个映射的元数据。其 Qwen3-8B 设置为温度 0.6、top-p 0.9，随机种子为 2026年1115。提案模型收到以下记录的提示词。

> 你将收到仅含校准数据的 JSON，其中包含规范单元 ID、形状、架构连通性、逐单元校准统计数据、量化约束、确切字节预算以及 48 个可行映射的元数据。返回一个 JSON 对象，其中包含 protocol_id 、 pool_id 、 task 、 block 、 status 、 required_w8、preferred_w4、n4_min、n4_max、min_hamming 和 candidate_ids。请从所提供的候选池中恰好选择 12 个唯一 ID。遵守每一项硬约束。不得使用开发数据或最终数据、分数、标签、奖励或未提供的信息。如果不存在合法子集，请返回 status="infeasible"以及被违反的约束 ID。不得放宽约束。

对于加载的权重张量$\mathbf{W}_i$，权重_最大_绝对值记录 最大 $|\mathbf{W}_i|$，且敏感性根据附录中估计器所述，在校准源上进行测量：D.1。连通性遵循捕获的执行图。对于每项测量，字段估计器用于指定计算该测量值所采用的规则。`cal_清单_sha256` 用于标识校准所用的输入。记录的有效响应返回状态=“ok”，且必需的 W8 列表和首选的 W4 列表均为空。它们报告 $n_4^{\min} = n_4^{\max} = N_4$、最小汉明距离 2，以及表S38中的任务专属 ID。Long 修复记录拒绝 C48 并使用 C44。Wall-L 修复记录将 C48 替换为 C46。这些记录保留协议/池/任务/区块标识符、被拒绝和已修复的响应、合法性/唯一性/预算验证、修复上限 2，以及回退为 false。两次修复均未成功后，将使用确定性的校准启发式方法。响应台账关联运行 ID、提示词、输入负载、原始响应、验证错误、修复、回退决策和所选候选 ID。第B.6节报告了向智能体提供的校准输入所发生的受控更改。

源 ID 用于标识观测及其原始父级回合或轨迹。该账本使用 source_id 表示观测，使用 parent_id 表示其父级。在同一父级内，child_index 用于标识子项。账本还在 split 中给出数据划分，在 block 中给出区块。键 raw_record_ref 用于定位原始记录，而 raw_content sha256 用于验证其内容。我们在 dataset_revision 中记录数据修订版本，并使用 session id 和 initial_

表 S84。分布偏移下的质量损失及重新选择带来的增益。

工作负载	原始冻结	分布外冻结	分布外开发集重新选择	重新选择增益	原始分布与分布外损失
Long	93.90	88.21	92.81	+4.60	5.69
TwoRoom	86.27	79.57	84.56	+4.99	6.70
Windows	37.70	30.55	36.45	+5.90	7.15

表 S85。对 LeWM Wall 的匹配干预效应（百分点）。

比较	区块效应（百分点）	
PTQ4WAM–FP32	48	+16.22
PTQ4WAM–匹配噪声	48	+14.38
PTQ4WAM–恢复敏感项	48	+11.39
PTQ4WAM–随机恢复	48	+4.00
PTQ4WAM–FGR 已禁用	48	+0.00
PTQ4WAM–精度置换	48	+10.51

state_id。最后两个字段分别给出随机种子和结果，即 seed 和 outcome。预先指定的父级/依赖组会在枚举其子观测之前被分配到一个数据划分和区块中。配对方法使用相同的父节点、输入和随机种子。规划源的所有重复项、Ranking回合中的比较，以及来自 Windows 轨迹的窗口，都会与其父级保持一致。例如，：Windows:final:0042:w2 是 :trajectory:0042 的子项 2，其同级索引 0–3 被分配到 FINAL 区块 04。源 ID 仅在标识完整父级本身时才等于其父级 ID。

表格S96 区分了每种协议中父源与其贡献的各个结果。

对于每种 S/N/A/F 工作负载和协议，每个区块包含 64 个校准父级、128 个开发父级和 256 个最终父级，十二个区块共计 5376 个父级。记录到的跨分割父级重叠、跨区块父级重叠以及未匹配方法对均为零。S/N/A 的每个父级有三个规划/排序结果，S/N 窗口的每条轨迹有四个窗口。F 的每个最终状态对应一个结果。子级计入其父级的评分。推断采用十二个配对区块内的差值。R 数据包含来自 180 个回合的 540 次比较。在 -J 中，每项任务在冻结选定映射前，校准和开发数据各使用一次。只有 FINAL 被划分为十二个区块，CAL/DEV/FINAL 父级来源彼此不重叠。在 -JR 中，12 次搜索各自独立抽取 64 个 CAL 、 128 个 DEV 和 256 个 FINAL 父级，各次运行之间不共享父级。-CB 复用 -JR 中的 Long 运行区块。E 执行协议保留自己的配对运行记录，不纳入这些校准/开发/最终计数。我们在单独字段中记录父级、内容摘要和初始状态重叠。运行记录还会注明实际的检查点/代码/内核修订版本、路径、形状、数据类型、组大小、批次、上下文、计时边界、分配量/已保留/进程字节数以及回退情况。每种协议都保留自己的观测数据，用于计算均值、方差和检验统计量。表 S97检查不同检查点和规划器随机种子下的 Wall 结果。

对于每个单元，我们抽取均值为零的噪声，并将其 Frobenius范数设为与对应的量化误差相等。敏感恢复和随机恢复会修改六个符合条件的单元。随机种子为 11、23、37、53 和 71。精度置换与参数数量和字节预算相匹配。预言机控制使用固定的动作候选项和 32 个展开种子。在报告综合严重失败率之前，分别统计不安全终止状态、碰撞和超时。在记录的 -F:Wall-L:01干预中，六个敏感恢复单元为 u003、u018、u037、u061、u074、u096。种子 11 对应的随机恢复单元为 u008、u022、u043、u069、u081、u102。噪声种子为 20261117，每个单元的噪声均值为零，且 Frobenius范数与该单元的量化误差相匹配。置换保持在等参数类别内。预言机展开使用种子 0–31。严重失败包括不安全结果、碰撞和超时。运行日志为 -F-Wall-L-01-log。

M. 配对质量与执行测量

S（任务与对角线）和 N（Agent-12 与广泛 48）协议在每个工作负载下均包含 12 个独立配对区块，随机种子为 20261115。这些记录和估计量与表 S75 中的 96 区块选择器均值、64 次运行执行均值、48 区块原生比较，以及由四项检验构成的 Holm 检验族保持分开。E（调优适配器与 PTQ4WAM）在每个工作负载下保留 12 个实测执行对。全部 20 个端点均值都与表 S66 中相应的 64 次运行均值一致；E 作为实测配对记录进行分析，不额外增加独立重复试验。–x 配对计时数组按照其自身协议进行分析（表 S19）。每个 S/N 区块包含 256 个最终源簇，每个规划/排序源有三个结果，或每条 Windows 轨迹有四个窗口。前七个工作负载的分母为 768，Windows 的分母为 1024。独立统计单位是配对区块，而非单个试验/窗口。表 S96 给出了父级统计。另有 A、F 和 R 三项研究，各自拥有独立的配对区块和检验系列。

对于配对差值 d_i，报告样本标准差 s_d 和逐点双侧 95%区间 $\bar{d} \pm 2.2009851601 s_d/\sqrt{12}$（$df = 11$）。S 使用 $d_i = 100\,(T_i - D_i)\,/m$，并对八个工作负载进行双侧配对-$t$ 检验，采用 Holm 校正。N 使用 $d_i = 100\,(A_i - B_i)\,/m$，并进行八项单侧非劣效性检验，检验对象为

表 S86. 每项检验包含 12 个区块的四项配对检验。

Test	对照组	处理组	增益 ±标准差	95% t 置信区间
A：长任务–对角线	92.111545 ± 0.692252	93.674045 ± 0.766227	1.562500 ± 0.415481	[1.298516, 1.826484]
字节：立方体任务–对角线	82.486979 ± 1.157317	84.006076 ± 1.360926	1.519097 ± 0.431257	[1.245090, 1.793105]
C：Long $q_{00} \to q_{11}$	8.382000 ± 0.196155	6.595000 ± 0.245981	1.787000 ± 0.113673	[1.714776, 1.859224]
D：Wall PTQ4WAM–FP32	78.027344 ± 3.369044	94.238281 ± 1.364649	16.210938 ± 2.561499	[14.583438, 17.838437]

A、B 和 D 给出以百分比表示的分数和以百分点表示的增益。C 给出以秒为单位的时间。每项双侧符号检验分别给出校正前的 $p = 0.00048828125$ 和经过四项配对检验的 Holm 校正后的 $p = 0.001953125$。

表S87。部署计数所隐含的符号检验界限。

工作负载	计数 N	PTQ4WAM–FP32计数变化	可实现的 $p_{\min}$
空间	500	-8	0.0078125
物体	500	-1	1
Goal	500	-4	0.125
Long	500	-10	0.001953125
Wall-L	50	8	0.0078125
Push-L	150	-2	0.5
TwoRoom	100	-1	1
Cube	50	3	0.25
Reacher	50	-1	1
Wall-D	50	2	0.5
PointMaze	50	0	1
Push-D	50	0	1
Meta-World Reach	50	-1	1

这些双侧精确配对符号界仅使用观测边际计数。

表 S88。双侧符号检验的不一致阈值。

系列 m	α	最少胜出次数	失败次数	p
1	0.05	6	0	0.03125
13	0.05	10	0	0.001953125
15	0.05	10	0	0.001953125
4	0.05	8	0	0.0078125

在 $\alpha = 0.05$，所有不一致配对都支持同一种方法。

表 S89。在失败概率为 5% 时的有限总体界限。

开发源	界限（百分点）
10	100.000
32	62.118
48	50.719
64	43.924
128	31.059
128 ($K = 48$)	34.370
512	15.529

所有行均使用 $K = 12$，除非标记为 $K = 48$。这些界限涉及采样误差，不衡量性能增益。

相对于 -0.50 个百分点。单侧 95% 同时下界采用 $\bar{d} - 2.9808724301 s_d/\sqrt{12}$。E 使用$d_i = T_{A,i} - T_{W,i}$ 或 $M_{A,i} - M_{W,i}$，其中时间和分配量-峰值端点分别归入 Holm 五重检验族。S/E 中每项任务的双侧区间并非同时区间。百分比和百分点差值在最终舍入前均使用整数计数计算。缩减采用测量值之比计算

表 S90。预训练模型的注册表命名空间。

系列	注册表参考值
Fast-WAM	/Fast-WAM/单元
LeWM	/LeWM/单元
DINO-WM	/DINO-WM/单元
V-JEPA 2-AC	/V-JEPA2-AC/单元

均值，而不是各区块比率的均值。表格 S98 和 S99总结了任务选择和候选名单非劣效性。表格 S100和 S101 给出了这些估计值背后的留出计数。表格S102 和 S103 展示了固定映射的节省量。相应的配对时间和峰值分配列于表格 S104 和 S105。

N. 范围与可复现性

制品。 计划发布代码。附录L列出了检查点和组件注册表、序列化字节、地图枚举，以及将运行与表格连接起来的标识符。协议各节给出了源清单、配对区块数组、调优与准备开销、统计分析和模拟器计时。附录L.3 记录了提案提示和修复流程。 10^{-8} 公式（S11）中的稳定项确保分母中的归一化为零时，误差仍为有限值。

伦理。 模型部署前，必须有人监督其使用，并测试其是否适用于预期用途

表 S91。全部 15 个工作负载的精确精度映射计数。

Task	N4/N	$N_{4,\mathrm{small}}/N_{4,\mathrm{large}}$	K	Pool	P_4	P_8	P_{16}	P_{32}
空间	614/614	410/204	1	1	5142216704	0	8388608	0
物体	614/614	410/204	1	1	5142216704	0	8388608	0
Goal	610/614	407/203	12	48	5112856576	29360128	8388608	0
Long	369/614	239/130	12	48	3183476736	1958739968	8388608	0
Wall-L	72/108	47/25	12	48	9633792	4521984	262144	0
Push-L	92/108	58/34	12	48	12713984	1441792	262144	0
TwoRoom	72/108	48/24	12	48	9437184	4718592	262144	0
Cube	100/108	65/35	12	48	13434880	720896	262144	0
Reacher	108/108	72/36	1	1	14155776	0	262144	0
Wall-D	66/66	44/22	1	1	77856768	0	1048576	0
PointMaze	66/66	44/22	1	1	77856768	0	1048576	0
Push-D	42/66	26/16	12	48	53084160	24772608	1048576	0
Meta-World Reach	72/72	48/24	1	1	84934656	0	1048576	0
排序	160/261	107/53	12	48	334495744	209715200	4194304	0
Windows	144/259	94/50	12	48	308281344	233832448	4194304	0

小单元和大单元分别包含 d^2 和 $4d^2$ 参数。映射标识符仅在候选池内有效。要跨协议匹配标识符，必须使用相同的注册表顺序和相同的映射摘要。

表 S92。序列化存储组件（字节）。

Task	W4 B	W8 B	比例 B	受保护的 B	标头 B	填充 B	$B = S$ 字节
空间	2571108352	0	321388544	16777216	9008	208	2909283328
物体	2571108352	0	321388544	16777216	9008	208	2909283328
Goal	2556428288	29360128	321388544	16777216	9008	208	2923963392
Long	1591738368	1958739968	321388544	16777216	9008	208	3888653312
Wall-L	4816896	4521984	884736	524288	4960	160	10753024
Push-L	6356992	1441792	884736	524288	4960	160	9212928
TwoRoom	4718592	4718592	884736	524288	4960	160	10851328
Cube	6717440	720896	884736	524288	4960	160	8852480
Reacher	7077888	0	884736	524288	4960	160	8492032
Wall-D	38928384	0	4866048	2097152	4624	240	45896448
PointMaze	38928384	0	4866048	2097152	4624	240	45896448
Push-D	26542080	24772608	4866048	2097152	4624	240	58282752
Meta-World Reach	42467328	0	5308416	2097152	4672	192	49877760
排序	167247872	209715200	34013184	8388608	6184	216	419371264
Windows	154140672	233832448	33882112	8388608	6168	232	430250240

对于所有工作负载，零点存储空间均为零。所显示的组件总和为 $B = S$。对于 Windows，表S91–S93 涵盖了候选包。报告的 5.078980× 比率以其 430.25 MB 为分母。首次独立部署还需要所有候选项共用的 1,542,173,696 字节固定 BF16 编码器权重。

。使用这些模型和数据时，也必须遵守其许可证和适用的隐私要求。

O. 讨论

固定精度下的可行性

一个精度映射可能在一条执行路径上满足设备预算，却在另一条路径上超出预算，即使两条路径返回相同的任务结果。对于工作负载 w，令 $\mathcal{F}_w$ 表示主论文式（2）中的可行映射–路径对。可接受的映射及每个映射对应的最快可行路径 are

$$\mathcal{C}_w^{\mathrm{adm}} = \{c : \exists r,\ (c,r) \in \mathcal{F}_w\}, \qquad r_c \in \operatorname*{arg\,min}_{r:(c,r)\in\mathcal{F}_w} L_{cr}. \tag{S47}$$

PTQ4WAM 使用完整模型测量值和数值检查，在分配开发评估之前确定这组路径。例如，在短上下文 DINO-WM Wall 上，打包路径和仅 FGR 路径均满足 5.10 秒延迟和 985 MiB 分配限制。在 5.50 秒和 840 MiB 的条件下，只有组合式 FGR+DHR 路径符合要求（表 S69）。

表 S93. 15 个工作负载的精度占比和制品大小。

Task	模块 W4 (%)	全参数 W4 (%)	W4 载荷 (%)	每个符合条件参数的字节数	制品 (MB)	FP32 载荷 / 制品
空间	100.00	99.8371	99.3517	0.565764	2909.283328	7.081614×
物体	100.00	99.8371	99.3517	0.565764	2909.283328	7.081614×
Goal	99.35	99.2671	98.2272	0.568619	2923.963392	7.046060×
Long	60.10	61.8078	44.6208	0.756221	3888.653312	5.298086×
Wall-L	66.67	66.8182	48.8372	0.759621	10.753024	5.363299×
Push-L	85.19	88.1818	76.3780	0.650825	9.212928	6.259864×
TwoRoom	66.67	65.4545	47.3684	0.766565	10.851328	5.314712×
Cube	92.59	93.1818	84.3621	0.625362	8.852480	6.514748×
Reacher	100.00	98.1818	93.1034	0.599899	8.492032	6.791270×
Wall-D	100.00	98.6711	94.8882	0.589499	45.896448	6.876815×
PointMaze	100.00	98.6711	94.8882	0.589499	45.896448	6.876815×
Push-D	63.64	67.2757	49.6933	0.748589	58.282752	5.415348×
Meta-World Reach	100.00	98.7805	95.2941	0.587249	49.877760	6.895517×
排序	61.30	60.9943	43.4014	0.770604	419.371264	5.230737×
Windows	55.60	56.4299	38.8889	0.793653	430.250240	5.078980×

表 S94。部署工件和结果标识符。

工作负载	工件 ID	结果 ID
空间	-R:空间:C00	-T1:空间
物体	-R:物体:C00	-T1:物体
Goal	-R:目标:C00	-T1:目标
Long	-R:Long:C00	-T1:Long
Wall-L	-R:Wall-L:C00	-T1:Wall-L
Push-L	-R:Push-L:C00	-T1:Push-L
TwoRoom	-R:TwoRoom:C00	-T1:TwoRoom
Cube	-R:Cube:C00	-T1:Cube
Reacher	-R:Reacher:C00	-T1:Reacher
Wall-D	-R:Wall-D:C00	-T1:Wall-D
PointMaze	-R:PointMaze:C00	-T1:PointMaze
Push-D	-R:Push-D:C00	-T1:Push-D
Meta-World Reach	-R:MetaWorld-Reach:C00	-T1:MetaWorld-Reach
排序	-R:排序:C00	-T1:排序
Windows	-R:Windows:C00	-T1:Windows

表 S95。注册表和父来源记录字段。

字段	记录值
记录类型/单位	测量单位记录 / .u000
模块路径	.编码器.区块0.q_proj.权重
形状 / 权重数量	[2048, 2048] / 4,194,304
最大绝对权重	1.384726
校准敏感度	0.00817362
父级 / 来源	-episode-001 / -episode-001:重复-2
数据划分 / 原始指纹	CAL / -parent-001
数据集/检查点/代码/内核修订版 -r1	

RoboCasa Windows 上的权重存储空间和进程内存

2,849 MB 的进程峰值包含在一块 RTX 4090 上进行的视觉编码和动作条件预测。进行推断时，facebookresearch/vjepa2 中的 vjepa2_ac_vit_giant 入口点会从 Meta 的 vjepa2-ac-vitg.pt 检查点加载编码器和预测器条目。这些模块共包含 1,317,394,944 参数（表 S106）。推断时会省略仅用于训练的目标编码器和优化器状态。

候选包占用 430,250,240 字节，包括量化权重、缩放因子、受保护张量、头部和填充（表 S92）。全部候选项还共享 1,542,173,696 字节的固定 BF16 编码器权重。因此，首次独立部署需要 1,972,423,936 字节的权重。一旦共享权重就位，

表 S96. 按协议划分的父级来源和最终结果。

协议 / 工作负载	区块	校准父级	开发父级	最终父级	每区块结果
S/N、规划或排序	12	768	1536	3072	768
S/N, Windows	12	768	1536	3072	1024
A、Long	12	768	1536	3072	768
F, Wall-L	12	768	1536	3072	256
R，排序	12	—	—	180集	45次比较
-J，规划	12	64	128	3072	768
−J, Windows	12	64	128	3072	1024
-JR，规划	12	768	1536	3072	768
−JR, Windows	12	768	1536	3072	1024
-CB，Long（JR匹配）	12	768	1536	3072	768

CAL、DEV 和 FINAL 在整个协议中统计父节点。在 -J 中，只有 FINAL 被划分为 12 个区块。

表 S97。LeWM Wall 按检查点和规划器种子统计的结果。

检查点	种子 17	种子 29	种子43
c1	199/241	204/238	197/242
c2	202/239	196/237	201/240
c3	198/243	205/239	200/242

表 S98. I　任务与代理指标选择的独立配对比较　　选择。

Task	计数/区块	对角线（%）	任务（%）	Δ±标准差（百分点）	95%置信区间（百分点）	Holm p
Long	768	91.5799	92.9579	1.3780 ± 0.5310	[1.0407, 1.7154]	4.24×10^{-6}
Wall-L	768	92.3503	94.0647	1.7144 ± 0.4471	[1.4304, 1.9985]	3.25×10^{-7}
Push-L	768	88.9648	90.4731	1.5082 ± 0.4281	[1.2362, 1.7803]	5.87×10^{-7}
TwoRoom	768	84.7765	86.6645	1.8880 ± 0.5043	[1.5676, 2.2084]	3.65×10^{-7}
Cube	768	82.8993	84.4293	1.5299 ± 0.4376	[1.2519, 1.8080]	5.87×10^{-7}
Push-D	768	91.1892	92.6107	1.4214 ± 0.6438	[1.0124, 1.8305]	9.99×10^{-6}
排序	768	89.7027	91.0590	1.3563 ± 0.4660	[1.0602, 1.6524]	2.73×10^{-6}
Windows	1024	35.6527	37.3617	1.7090 ± 0.6169	[1.3170, 2.1010]	3.34×10^{-6}

区间为逐点 95%，p 值采用 Holm 校正（针对八项双侧检验）。

表 S99. 智能体候选名单相对于完整样本池的非劣效性。

Task	计数/区块	广泛（%）	智能体（%）	Δ±标准差（百分点）	L (百分点)	Holm 非劣效性 p
Long	768	94.1406	94.1081	-0.0326 ± 0.2428	-0.2415	5.83×10^{-5}
Wall-L	768	95.5838	95.4970	-0.0868 ± 0.1504	-0.2162	4.82×10^{-6}
Push-L	768	91.5256	91.4171	-0.1085 ± 0.2653	-0.3368	0.000169
TwoRoom	768	86.9141	86.8273	-0.0868 ± 0.1950	-0.2546	3.66×10^{-5}
Cube	768	85.0911	84.9501	-0.1411 ± 0.2183	-0.3289	0.000139
Push-D	768	92.1441	92.0573	-0.0868 ± 0.2102	-0.2677	5.83×10^{-5}
排序	768	92.5456	92.5998	0.0543 ± 0.2183	-0.1336	9.18×10^{-6}
Windows	1024	39.1439	39.0462	-0.0977 ± 0.1815	-0.2538	2.89×10^{-5}

同时进行的单侧　　在全部八种工作负载上，95%下界均超过 -0.50 个百分点的幅度。这八项单侧检验采用 Holm 校正　　t.

切换映射只需替换候选包。

在 Windows回放05 中，内存轨迹跟踪了同一个进程 ID，从权重加载开始，直到 51.779 秒重放结束（表 S18）。注册表、有效位集和包摘要将该轨迹与表 S91 和 S92关联起来。该进程以批大小为 1、历史长度为 64 的配置，接收原始 RGB 观测、机器人状态和动作。经过 20 次预热调用后，我们对 100 次同步调用进行了计时。每次调用都会对当前观测进行编码，并运行条件预测器。同一进程在监控下对初始历史记录进行编码，并将其特征保留在 GPU 内存中。因此，整个重放过程都涵盖了从原始输入开始的视觉编码，不使用离线特征，也不涉及辅助编码器进程。

表S16在三个嵌套作用域报告峰值内存。已分配张量达到 2,206 MB，分配器已保留

表 S100。12 个配对区块中，对角 Hessian 和任务选择的留出计数。

	Long		Wall-L		Push-L		TwoRoom		Cube		Push-D		排序		Windows	
区块	D	T	D	T	D	T	D	T	D	T	D	T	D	T	D	T
01	708	717	703	722	687	695	638	656	622	632	706	720	677	685	371	393
02	706	717	712	727	683	693	656	665	654	664	686	701	677	692	368	386
03	697	703	712	728	685	700	636	654	625	630	695	697	704	713	364	377
04	704	720	720	732	681	697	647	660	638	648	704	715	688	700	358	380
05	702	715	689	703	696	713	664	677	640	654	698	706	688	699	379	399
06	706	715	705	715	659	671	661	672	624	637	694	702	683	694	364	368
07	703	714	713	730	691	704	652	665	642	652	701	709	710	712	376	387
08	710	715	713	726	681	690	643	654	653	670	696	709	683	693	364	381
09	699	708	715	724	688	699	655	676	621	633	702	718	694	707	355	384
10	704	722	710	718	683	693	657	677	646	656	702	722	687	694	350	366
11	696	702	706	716	677	683	636	648	635	652	701	711	688	701	358	375
12	705	719	713	728	688	700	668	683	640	653	719	725	688	702	374	395

D 表示对角海森矩阵选择，T 表示任务选择。两种方法在每个工作负载中共享最终父级。

表 S101. 12 个配对区块中完整提议池和受限提议池的留出计数。

	Long		Wall-L		Push-L		TwoRoom		Cube		Push-D		排序		Windows	
区块	B	A	B	A	B	A	B	A	B	A	B	A	B	A	B	A
01	731	730	726	727	705	704	684	684	635	635	706	707	710	710	412	410
02	720	717	740	740	695	695	654	653	655	656	708	707	694	694	371	370
03	721	722	729	729	713	712	669	669	638	637	695	694	722	722	423	420
04	714	716	730	729	707	706	666	668	661	659	710	711	704	704	400	398
05	720	720	733	730	698	696	655	653	654	652	712	710	708	707	404	404
06	736	734	723	723	697	699	671	671	648	647	712	708	728	726	391	392
07	722	722	740	738	704	702	672	669	662	660	712	710	711	710	401	400
08	724	721	738	738	691	688	664	665	654	653	722	721	704	706	406	403
09	726	725	740	740	713	711	663	660	659	655	692	694	715	717	403	405
10	715	715	738	738	709	706	672	671	665	665	714	714	715	714	385	387
11	713	714	732	730	712	711	664	664	648	645	709	709	700	703	400	398
12	734	737	740	739	691	695	676	675	663	665	700	699	718	721	414	411

B 表示 Broad-48，A 表示 Agent-12。独立的 N 和 S 协议使用不同的源清单。

表 S102. 12 次配对运行中的固定映射重放节省（秒）。

Task	ΔT ± 标准差（秒）	95% 置信区间（秒）	时间缩减	Holm p
Push-L	0.2980 ± 0.0472	[0.2680, 0.3280]	8.19%	4.12×10^{-10}
Cube	0.4720 ± 0.0606	[0.4335, 0.5105]	10.09%	6.31×10^{-11}
Long	0.9350 ± 0.0865	[0.8800, 0.9900]	12.42%	2.98×10^{-12}
Wall-D	0.4280 ± 0.0808	[0.3766, 0.4794]	7.81%	1.35×10^{-9}
PointMaze	0.7840 ± 0.0753	[0.7361, 0.8319]	11.55%	3.6×10^{-12}

五项配对时间比较的逐点95%区间及 Holm 校正。

表S103. 12 次配对运行中的固定映射分配节省（MB）。

Task	ΔM ± 标准差（MB）	95% 置信区间（MB）	峰值减少	Holm p
Push-L	9.00 ± 0.53	[8.66, 9.34]	5.13%	1.22×10^{-14}
Cube	12.50 ± 1.27	[11.69, 13.31]	7.08%	1.68×10^{-12}
Long	703.20 ± 25.69	[686.88, 719.52]	9.00%	1.12×10^{-16}
Wall-D	54.70 ± 3.20	[52.67, 56.73]	5.89%	1.22×10^{-14}
PointMaze	86.80 ± 4.96	[83.65, 89.95]	8.60%	1.22×10^{-14}

逐点95%区间以及一个单独的 H 用于五项内存比较的 Holm 系列。任务结果 在每个源上均一致。

高达 2,416 MB，而驱动程序报告的进程峰值为 2,849 MB。距离 3,100-MB 的进程限制还剩 251 MB。在权重、缩放因子和输入固定的情况下，调优适配器在每个窗口上都与 PTQ4WAM 的结果一致

但最终达到 3,388 MB。更改路径节省了 539 MB，使映射符合预算。完整的 100次调用重放的 p95时长也从 58.405 秒降至 51.779 秒（表 S19）。

表 S104. 按工作负载划分的配对适配器与 PTQ4WAM 重放时间（秒）。

	Push-L		Cube		Long		Wall-D		PointMaze	
区块	适配器	Ours	适配器	Ours	适配器	Ours	适配器	Ours	适配器	Ours
01	3.6809	3.4286	4.6316	4.2085	7.7706	6.7516	5.5174	5.1124	6.8556	6.1278
02	3.5952	3.3556	4.5329	3.9798	7.8268	6.9477	5.4165	5.1012	6.5223	5.5464
03	3.5815	3.2859	4.4127	3.8173	7.7269	6.8117	5.4869	4.9306	6.8914	6.0677
04	3.5650	3.2613	4.7131	4.2645	7.4221	6.4111	5.6255	5.1735	6.9430	6.1000
05	3.3936	3.0648	4.8275	4.3864	7.6205	6.6800	5.4724	5.2024	6.7890	5.9885
06	3.7141	3.3623	4.6847	4.2235	7.4366	6.5840	5.5611	5.1482	6.8945	6.1055
07	3.7040	3.4399	4.6830	4.2284	7.3528	6.4368	5.3173	4.8889	6.7323	6.0056
08	3.8337	3.4582	4.8555	4.3925	7.1811	6.3279	5.3685	4.9835	6.7573	6.0607
09	3.7182	3.3628	4.5420	4.1365	7.7976	6.8564	5.4985	5.0299	6.6800	5.9398
10	3.6672	3.4363	4.7782	4.3510	7.4445	6.6423	5.4839	4.9786	7.0830	6.2911
11	3.6577	3.3632	4.7214	4.1682	7.4877	6.5143	5.5044	4.9931	6.7640	5.9917
12	3.5689	3.2851	4.7774	4.3394	7.2928	6.1762	5.5076	5.0817	6.5676	5.8472

表 S105. 按工作负载列出的配对适配器和 PTQ4WAM 峰值分配量（十进制 MB）。

	Push-L		Cube		Long		Wall-D		PointMaze	
区块	适配器	Ours	适配器	Ours	适配器	Ours	适配器	Ours	适配器	Ours
01	175.47	166.57	176.09	163.00	7761.63	7040.33	934.70	882.64	1010.88	919.59
02	175.65	166.47	175.98	164.04	7879.82	7223.60	926.06	875.65	1006.21	923.92
03	175.36	166.46	175.68	162.33	7775.73	7075.95	928.88	869.51	1008.68	916.37
04	175.29	167.06	177.73	167.23	7792.33	7109.15	922.50	869.62	1006.80	917.82
05	175.44	166.40	176.35	164.03	7817.69	7129.25	921.75	871.82	1009.35	915.20
06	174.83	164.95	176.14	163.09	7825.68	7087.22	928.50	870.43	1011.44	931.04
07	175.92	166.73	177.55	164.14	7809.40	7070.66	934.59	878.53	1007.79	923.63
08	175.63	166.98	176.48	164.24	7812.55	7110.69	927.00	871.04	1011.08	923.79
09	176.16	168.06	176.34	166.13	7839.90	7134.51	924.65	867.86	1014.40	928.39
10	174.14	165.36	176.78	161.86	7812.08	7142.03	925.85	867.46	1006.80	929.12
11	175.21	165.66	176.61	163.93	7797.90	7074.24	936.81	882.55	1013.69	923.92
12	175.70	166.10	176.27	163.98	7855.29	7143.97	930.71	878.49	1000.88	913.61

表 S106。按组件划分的窗口推理参数。前三行细分了表 S91中的窗口计数。共享的 BF16 权重单独存储。

权重	视觉编码器	预测器	总计
W4	140,509,184	167,772,160	308,281,344
W8	99,614,720	134,217,728	233,832,448
受保护	963,200	3,231,104	4,194,304
共享 BF16	771,086,848	0	771,086,848
总计	1,012,173,952	305,220,992	1,317,394,944

Windows 上的端点评分和源独立性

部署面板包含来自 14 条轨迹的每条轨迹中记录的四个窗口。FP32 在 56 个窗口中通过了 20 个，PTQ4WAM 则通过了 21 个（表 S46）。窗口 `capture-017/w2` 展示了预测如何越过预先设定的阈值。两个模型都接收了帧 96–103 的 RGB 帧、帧 103 时的机器人状态，以及引导至帧 111 的八个记录动作。两者都预测了帧 111 的视觉令牌。只有评分器能够访问观测到的端点。评分器复制帧 111，组成双帧输入，并使用相同的冻结上游 FP32 编码器生成通用参考值 z^{*}，其包含 256 个维度为 1,408 的词元。

评分器按照相同的 token 顺序将预测和参考值转换为 FP32，并累加逐元素绝对差值，累加过程使用 FP64。当窗口内所有 $256 \times 1,408 = 360,448$ 个值的均值不超过 0.100 时，该窗口即通过；这是在 CAL 上固定的容差。FP32 预测的绝对误差总和为 37,486.592，均值为 0.104，得分为 0。对于 PTQ4WAM，总和为 34,603.008，均值为 0.096，得分为 1。两项得分都可以根据三个形状相同的数组重新计算：pred_fp32、pred_ptq 和参考值。

这项独立的 96 区块选择研究使用了冻结的内部 RoboCasa Windows v3 捕获快照。我们通过每个父项原始的捕获会话 UUID、原始片段 UUID，以及窗口化之前状态和动作流的摘要来识别父项。多个相机导出文件、重新编码的视频以及同一回合的重复导入共享一个父级键。我们在提取每个父项的四个窗口之前，将其分配到一个数据划分和一个区块中。账本记录了 6,144 个 CAL 父项、12,288 个 DEV 父项和 24,576 个 FINAL 父项，总计 43,008 个。审计发现，不同数据划分之间没有共享的原始捕获，不同区块之间也没有共享的原始捕获。

相对于可行解池的反馈成本

每次独立的 `-JR` 搜索都会在表 S7所列的存储空间、进程内存和单次调用 p95 限值下，对48个映射中的每个映射搜索两条路径。$F = |\mathcal{C}_w^{\mathrm{adm}}|$ 表示该次运行的值。拥有两条可行路径的映射只计一次，并保留其

最快的可行路径。两种选择策略使用相同的配置文件。对可行解池进行穷举评分需要 $128F$次候选项–开发集父项访问。对候选名单评分需要 $128\times12=1,536$ 次访问，节省比例为 $1-12/F$。Pareto-48 对全部48个映射评分，并选出最佳可行项。在相同开发分数和分层规则下，仅对可行映射进行穷举评分会得到相同的胜出项，因此其留出得分可直接从表S8中读取。

在 Windows 上，12 次运行的可行方案计数总和为 288（表S107）。穷举评分需要$288\times128=36,864$ 次访问。候选名单使用了 18,432 次访问（减少 50.00%），在留出窗口上的评分为 39.95%，而可行解池为 40.08%。在四种工作负载中，筛选将性能分析后的访问次数从 176,640 降至 73,728，节省了 102,912 次访问（58.26%）。按工作负载计算，节省幅度为 50.00% 至 63.64%；留出集评分比穷举可行解池评分低 0.07–0.14 个百分点。主论文中的 75% 节省率以固定的 48 映射审计为分母。候选名单相对于 Pareto-48 的四个同时单侧下界均超过预先设定的 -0.50 个百分点非劣效性界值（表S8）。

质量优先对照会先对提案评分，再强制满足可行性。在约束贝叶斯优化（BO）中，会利用已完成的开发集得分来选择后续可行提案。在候选池、配置文件、上限、评分规则和 12 个分数的预算均相同的情况下，PTQ4WAM 在四种工作负载上的留出得分比约束 BO 高出 2.02–2.80 个百分点。在八项经 Holm 校正的比较中，四个同时置信下界均为正（见表 S8–S11）。

两种搜索协议下的准备时间

在 Fast-WAM Long 上，表 S31 报告了 96 个区块中每个选择区块的平均耗时为 79.4 秒，而 Broad-48 为 264.0 秒。该协议会重复利用已编译的路径和候选资源配置文件，然后通过并行滚动执行，为每个区块获取新的反馈。根据表 S41中的计时类别，Long 记录将 4.8 秒用于共享准备，1.5 秒用于提案，13.1 秒用于构建和打包，另有 60.0 秒墙钟时间用于并行开发评估。这些组件的总耗时为 79.4 秒。

表S43 报告称，12次独立完整搜索的平均耗时为8,275.1917秒。其PTQ4WAM-full行采用Long $-\mathrm{JR}$ 运行。每次运行都会对全部48个映射的两条路径进行性能分析，每条路径预热20次并计时调用100次，因此共有$48\times2\times120=11,520$次性能分析调用。随后，它在每个开发父项上评估12个候选项，共进行1,536次全时域访问。准备工作包括加载、编译与调优、调度，以及失败尝试和修复。PTQ4WAM-stage会在连续阶段复用已完成的候选项—父项结果，总共只需新增512次访问。其平均准备时间为3,495.8167秒，减少了57.8%，留出成功率则下降了0.23个百分点。在共享的4,000秒准备时间上限下，分阶段PTQ4WAM和分阶段BO的成功率分别为94.94%和92.87%，平均成本分别为3,495.8秒和3,503.9秒（表S43–S45）。

评分规则和提议器的影响

固定候选池后，改变用于选择胜出映射的规则，即可单独考察任务反馈的影响。在六项规划任务中，按任务成功率选出的映射，相较于按重建误差、迹海森矩阵评分和对角海森评分选出的映射，留出得分分别高出 3.66、2.83 和 1.52 个百分点（表 S30）。在针对八种工作负载开展的独立配对比较中，相较于按对角海森排序，优势介于 1.36 至 1.89 个百分点之间（表S98 和S100）。表S64 还列出了额外的重建和曲率控制项。

固定任务评分器和 12 次评估预算后，即可单独考察提议器的影响。六项规划任务的留出成功率均值分别为：Agent-12 为 90.93%，Sensitivity-12 为 90.37%，Fixed-12 为 90.15%，Random-12 为 89.70%（表 S40）。Agent-12 比同等预算下表现最佳的替代方案 Sensitivity-12 高出 0.56 个百分点。它仅使用校准数据和合规的候选元数据。其候选名单在开发评分前即已冻结，并接受相同的合法性和字节预算检查。Broad-48 使用 48 次评估，达到 91.00%。在针对八种工作负载的独立分析中，Agent-12 与 Broad-48 的所有同时单侧下界均高于 -0.50个百分点（表 S99）。上文的 Windows 适配器比较单独考察了第三项干预：在权重、缩放因子和输入固定的情况下改变执行路径。

LeWM Wall 上的动作选择

经过调优的 FP32 规划器和 PTQ4WAM 均接受了 12 次开发配置评估。在 12 个配对区块中，每个区块包含 256 个留出状态，两者的成功率分别为 83.1380% 和 94.2057%。配对增益为 11.0677 个百分点（95%置信区间 $[10.2805, 11.8549]$点，表 S34–S35）。

来自一个 Wall 初始状态的重放展示了预测成本较低的一方如何从一个动作序列切换到另一个。在候选集和规划随机种子相同的情况下，FP32 规划器为序列 A 和 B 分别分配了 0.412 和 0.426 的成本，因此选择了 A。选定的量化映射为 A 和 B 分别分配了 0.419 和 0.405 的成本，因此选择了 B。我们固定这两个序列，并在相同的 32 个环境随机种子上分别重放。A 在 13 个随机种子上成功，B 则在 26 个随机种子上成功。另一项匹配噪声对照也显示，平均特征误差作为动作质量预测指标有其局限。在报告的标准化误差均为 0.0225 时，PTQ4WAM 的首动作一致率为 93.4%，匹配噪声为 81.9%；两者的成功率分别为 94.25% 和 79.87%（正文表 2）。

表 S107。每次独立搜索将 12 个可行映射列入候选名单所节省的任务反馈。可行映射计数遵循运行顺序。节省量依据每种工作负载的 12 次运行中的总访问次数计算。配对留出差异见表 S8。

	每次运行的可行映射数（F）		节省的访问次数		留出得分（%）		配对差异
工作负载	运行 01–06	运行 07–12	计数	%	候选名单	可行解池	(pp)
Fast-WAM Long	32, 36, 29, 34, 31, 33	35, 30, 32, 34, 36, 34	32,256	63.64	95.1714	95.2365	−0.0651
LeWM Push-T	31, 27, 32, 29, 34, 30	28, 31, 29, 32, 27, 30	27,648	60.00	91.5582	91.6992	−0.1411
DINO-WM Push-T	27, 29, 25, 31, 28, 26	30, 28, 27, 29, 30, 26	24,576	57.14	93.6415	93.7500	−0.1085
RoboCasa Windows	23, 27, 21, 25, 22, 24	26, 23, 24, 28, 21, 24	18,432	50.00	39.9495	40.0798	−0.1302

Fast-WAM Long 权重与重放边界

Fast-WAM 将 Wan2.2-5B 视频骨干网络与一个约有十亿个参数的动作专家相结合。Long 注册表将其检查点标识为`-Fast-WAM-r1`。表S108给出了公开的 LIBERO Base 文件及其摘要。

Long 报告的 51.51 亿个参数，是指每个候选包中存储的参数。其中包括 3,183,476,736 个 W4 参数、1,958,739,968 个 W8 参数，以及 8,388,608 个受保护的 16 位参数（表 S91）。缩放因子、头部和对齐填充使候选包大小增至 3,888,653,312 字节（表 S92）。两个专家还使用 BF16 权重，这些权重仅存储一次，并由所有候选项共享（表 S108）。首次加载两个专家时会读取 5,628,744,704 字节，其中包括 1,740,091,392 字节的共享权重。

表 S108：Fast-WAM Long 按专家划分的参数。总计数中每个存储的参数仅计数一次。

存储空间	视频专家	动作专家	总计
候选项	4,368,367,616	782,237,696	5,150,605,312
共享 BF16	629,145,600	240,900,096	870,045,696
总计	4,997,513,216	1,023,137,792	6,020,651,008

LIBERO Base 发布版 `yuanty/fastwam`，修订版本 `b10939c`。文件 `libero_uncond_2cam224.pt`

SHA-256`1000437cfcf55c000094f79a2600634c502bcb5b492476b94bf8509883a49579`

一个图形处理器进程会同时持有两组权重，其内存峰值也包含这两组权重。固定重放从编码后的观测和指令开始。每次计时调用涵盖两个专家执行的算子。VAE 和文本编码在计时区间之前进行，而加载和编译计入准备阶段。正文表 1 中的 6.608 GiB 是重放期间测得的张量分配峰值。表 S7 中的 9,000 MB 是整个图形处理器进程的内存上限。

上游视频专家和动作专家的隐藏宽度分别为 3,072 和 1,024。Long 注册表条目 .u000 映射到 .encoder.block0.q_proj. 权重和图节点 .matmul000。其源是视频块 0 中的自注意力查询矩阵（q.weight）。打包器提取行和列 $[0, 2048)$，并将生成的 2048×2048 BF16 块连续存储在元素偏移量 $[0, 4194304)$。

Long 选择研究中的初始状态

这项 96 区块研究保留了 LIBERO Long 的十项任务定义。对于每个父级，我们重置模拟器并采样一个有效的物体布局。为确定一个物理初始状态，我们对任务定义、场景资源版本以及机器人和物体的完整初始状态进行哈希处理。哈希内容不包括父级 ID、时间戳和执行随机种子。无论按父级 ID 还是状态摘要计数，均得到 6,144 个 CAL、12,288 个 DEV 和 24,576 个 FINAL 状态。各划分之间以及各区块之间，父级 ID 和状态摘要均无重叠。因此，这 43,008 个不同状态描述了在这十项任务定义下采样的扩展分布。

例如，Long/B001/FINAL/FWL-E-000001 贡献了一个初始状态。它的三次执行（$r_0 = 1$、$r_1 = 1$、$r_2 = 0$）得分为 $2/3$。独立的完备搜索协议有自己的清单。其记录 -JR:Long:01:FINAL:p0000:r2 将结果 0 与 -state-Long-01-FINAL-0000 关联起来。43,008 这个计数仅来自 96 区块清单。

开发获胜者与部署标识符

15 个部署工作负载中有 9 个拥有多个候选项。Long 和 Wall-L 的 DEV 获胜者 `:S:Long:01:C20` 和 `:S:Wall-L:01:C34`分别以 `-R:Long:C00` 和`-R:Wall-L:C00`注册用于部署。按照表 S39 中的规定，对它们的有效位集进行哈希处理，得到的 SHA-256 前缀分别为 `74d0f77c6bea` 和 `a170dd39cba9`。附录 L.2 中的标识符排列保留了其余七个工作负载的原始编号（表S94）。对于这七个工作负载，锚点 C00 在 DEV 之前固定下来，并对 48 映射池中的候选项 C00–C11 进行评分。C00 在每项比较中均以一或两个 DEV 结果胜出（表 S109）。

正文表 1 中的 Long 部署使用了 C20，其别名为 C00。采用 FP32 时，固定重放时间为 13.17 秒；采用 PTQ4WAM 时则降至 6.57 秒（$2.00\times$）。在另一项为期 500 回合的部署评估中，FP32 在 477 回合中成功（95.4%），PTQ4WAM 则在 467 回合中成功（93.4%）（表S46）。上游 LIBERO 评估对每个套件中的十项任务分别使用 50 回合。在正文表 4 的独立搜索运行 01 中，PTQ4WAM 在路径 `-r02`上选择了 C17。在相同的 256 个留出父节点上，C17 在 768 次执行中成功了 737 次，而受约束贝叶斯优化选出的映射在 768 次执行中成功了 717 次（表S10）。

表 S109。保留原始 C00 的七个工作负载的开发结果。计数表示成功的 DEV 结果数占所有评估结果数的比例。摘要标识了按规范单元顺序排列的已部署映射。

工作负载	C00 得分	亚军	亚军得分	部署 ID	W4/单位	SHA-256 前缀
Fast-WAM 目标	124/128	C04	123/128	-R:目标:C00	610/614	042650fc5e9d
LeWM Push-T	350/384	C07	348/384	-R:Push-L:C00	92/108	6488384dd41f
LeWM TwoRoom	333/384	C03	332/384	-R:TwoRoom:C00	72/108	ab7d63c37f31
LeWM Cube	325/384	C09	323/384	-R:Cube:C00	100/108	95e8a7a80c09
DINO-WM Push-T	356/384	C05	354/384	-R:Push-D:C00	42/66	f867e5cc1488
Franka Ranking	352/384	C08	350/384	-R:排序:C00	160/261	dd9bcd17bf7e
RoboCasa Windows	198/512	C02	196/512	-R:Windows:C00	144/259	dc665268bd58

资源限制下的首次尝试成功率

在 Fast-WAM Long 上，采用 PTQ4WAM 时，95.17% 的留出执行在首次尝试时成功，且未发生资源回退；采用受约束贝叶斯优化时为 93.13%，采用质量优先选择（Late）时为 90.81%。两者分别提升了 2.04 和 4.36 个百分点。对正文表 4 重新评分时，每种方法和工作负载仍使用相同的 3,072 个留出父节点。如果某个父节点触发回退，其所有结果在首次尝试时的得分均为零。对于 Long，Late 最终成功的 8,606 次中有 237 次是在回退后取得的，因此为 $(8,606 - 237)/9{,}216 = 90.81\%$。回退频率按父节点计数，而成功率则按执行次数或窗口计数（表 S10–S12）。

在全部四种工作负载上，穷举Pareto-48的首次尝试成功率最高，其次是 PTQ4WAM、受约束贝叶斯优化和 Late（表 S110）。在使用 12 次任务评估的方法中，PTQ4WAM排名第一。对于这些相互独立的搜索，p95限值适用于每次在线决策中的一次模型调用。按表 S110中的工作负载顺序，限值分别为 85、45、280 和 680 毫秒（表 S7）。对于冻结搜索，按相同顺序，完整100次调用重放的 p95 限值分别为 7.60、4.00、24.00 和 61.00 秒（表S4）。

与先前工作的关系及测量范围

硬件感知精度选择（HAQ）、动作感知WAM量化（SteerQuant、QuantWAMs 和 Q-WAM）、低比特融合（Rudder 和 Marlin），以及非连续注意力访问为设计的不同部分提供了依据。Rudder 将缩放与补偿融合，用于动作引导的量化。在固定编码和缩放因子的情况下，FGR 移除了物化的组重缩放中间结果，DHR 则移除了兼容的视觉历史拼接。在相同的支持格式、存储目标和 12 次评估预算下，PTQ4WAM 在 Fast-WAM Long 上达到 95.37%，在 Goal 上达到 97.43%；而最强匹配适配方案分别为 93.16% 和 95.98%（表 S13）。在组大小为 128 的匹配 W4A16 内核实验中，原生 Marlin 耗时 7.103 秒，进程内存为 8,546 MB；而所评估的 PTQ4WAM 路径耗时 6.514 秒，进程内存为 8,022 MB（表 S14）。

在五种工作负载中，联合 FGR+DHR 将完整运行时间缩短了 7.81–12.42%，并将峰值分配内存减少了 5.13–9.00%（相较于调优后的同映射适配器；表 S66 以及表 S102–S105）。四路路径对照分别隔离了生产端组重缩放和直接历史读取（表 S20 和S56）。FGR 要求代码、缩放因子、分组和有效累加均保持一致。DHR 要求历史片段不可变，且逻辑位置、掩码和归一化保持不变（附录 J.5–J.6）。不支持的站点保留回退路径。全部 448 个数值案例均满足指定的 FP32/FP16 容差（表 S62）。附录 L–M 中的清单文件、位集、运行标识符和配对测量值，将这些检查与其评估协议关联起来。

在 15 个工作负载中，9 个允许多种合法的精度映射，6 个则在固定格式下测试执行。表 S46给出了任务特定的分子和分母，包括 Ranking 和 Windows 按来源分组的结果。归一化中位数恢复率概括了这些不同的端点。资源可行性是指在规定的进程配额和延迟限制下，对 RTX 4090 进行测量所得的结果。对于重复部署，通过单独计时的模拟器协议测得，使用冻结的分阶段制品时，每个包含 500 回合的 Long 批次耗时 675.7 秒，使用优化的 BF16 时耗时 973.0 秒。计入各自 3,469.4 秒和 48.6 秒的准备成本后，从第 12 个批次开始，累计耗时以 PTQ4WAM 为优（表 S71–S72）。

表 S110。正文表 4 中留出父节点的首次尝试成功率。每种规划工作负载包含 9,216 次执行，Windows 包含 12,288 个窗口。BO、PTQ4WAM 和 Pareto-48 均无回退，因此其首次尝试成功率与最终成功率一致。增益为 PTQ4WAM 减去所示方法的结果，单位为百分点。

	首次尝试成功率（%）				Late		增益（百分点）	
工作负载	Late	BO	PTQ4WAM	Pareto-48	回退父项 (%)	FINAL (%)	相较于 Late	相较于贝叶斯优化
Fast-WAM Long	90.81	93.13	95.17	95.24	84/3,072 (2.73)	93.38	4.36	2.04
LeWM Push-T	88.36	88.77	91.56	91.70	49/3,072 (1.60)	89.76	3.20	2.79
DINO-WM Push-T	90.72	91.62	93.64	93.75	40/3,072 (1.30)	91.88	2.92	2.02
RoboCasa Windows	36.81	37.15	39.95	40.08	58/3,072 (1.89)	37.52	3.14	2.80