QUALITY-AWARE ONLINE BUDGET CONTROL FOR LLM-BASED MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Cost-efficient language-model multi-agent systems (MAS) often target average cost reduction, whereas deployment under user quotas requires maximizing quality within a hard per-request budget. We propose Quality-Aware Online Budget Control (QOBC), a unified controller for single-round MAS with explicit dependencies and enforceable output limits. QOBC learns agent budget–quality curves through single-agent interventions, optionally selects among designer-approved execution graphs, and reallocates output budgets during execution. A shared solver accounts for generation and downstream input costs. Actual usage drives replanning; joint batch reservations enforce safety under explicit accounting assumptions. On 50 held-out MATH questions per controlled workflow, QOBC improves normalized budget–accuracy AUC over online equal allocation by 0.220 and 0.230 in sequential and parallel structures. On 100 new held-out inputs per native application, MoA AUC rises from 0.538 to 0.618 and Corex Review AUC from 0.570 to 0.739 relative to static quality allocation. On Screenplay, AUC rises from at most 0.292 with online equal allocation to at least 0.469, with the best quality at the 100% reference-budget level. No request-budget violation is observed across 9,000 native conditions spanning three applications.

1 INTRODUCTION

Language-model multi-agent systems (MAS) coordinate specialized agents for decomposition, reasoning, and verification (Wu et al., 2024; Hong et al., 2024; Qian et al., 2024). Collaboration can improve performance but increases token consumption through model calls and inter-agent communication. Under task-specific quotas, deployment therefore requires maximizing attainable quality within a given budget, rather than merely minimizing execution cost.

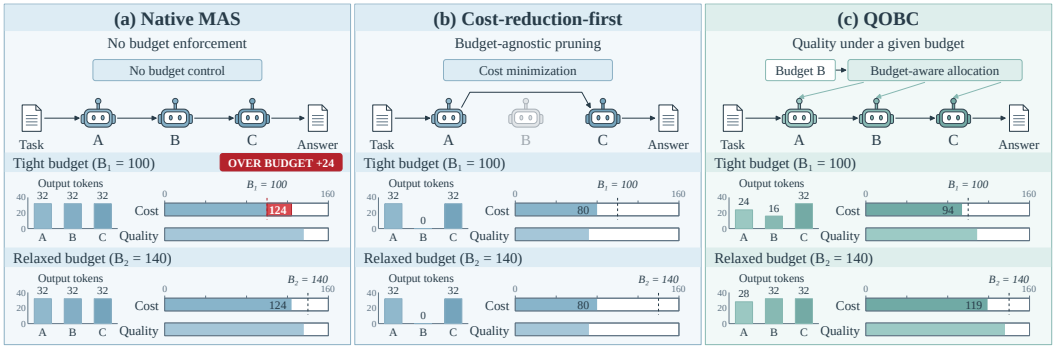


Figure 1: Budget control for the same three-agent workflow. Native execution can overspend; fixed pruning may leave capacity unused; QOBC allocates output limits within B. Bars are schematic, not experimental results. Cost assumes 16 fixed input tokens per active agent, only the preceding output forwarded, full use of output limits, and input/output prices of $0.25/1$; quality is illustrative.

Pruning redundant communication or agent calls reduces inference expenditure (Zhang et al., 2025b; Wang et al., 2025b), but lower average cost alone does not determine how an existing MAS should

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

allocate a tighter or more generous request budget. We study this allocation problem, seeking to preserve task quality at each affordable level (Figure 1).

An agent's marginal quality benefit depends on its role and graph position. Upstream outputs affect downstream reasoning and become paid inputs, so their cost extends beyond generation. Actual outputs may also be shorter than their limits, changing both the remaining budget and subsequent reading costs. Allocation must therefore account for communication and adapt to observed usage.

QOBC learns agent budget–quality curves and reallocates unfinished-agent limits after each batch settles. When a system supplies valid alternative graphs, it also selects one before execution. A shared KKT-based solver accounts for generation and downstream input costs; conservative input bounds and enforced output limits protect each dispatch. A common adapter supplies native prompts, dependencies, and any admissible alternatives. The selected graph stays fixed during execution, while online updates revise unfinished output limits and cost forecasts.

We evaluate native MoA, CrewAI Screenplay Writer, and Corex Review while preserving their call order and prompts under substituted model backends. Controlled sequential and parallel structures isolate allocation, graph choice, and replanning. QOBC improves budget–quality AUC over equal allocation in both controlled settings and over static quality allocation in all three native applications. Screenplay leads at the 100% reference-budget level, and no request-budget violation occurs in 9,000 native conditions under the stated accounting assumptions.

2 Related Work

MAS workflows. Single-round MAS traverse each participating agent once; multi-round protocols revisit agents. Either can use fixed or adaptive topology. AutoGen supports multiple protocols (Wu et al., 2024), whereas GPTSwarm optimizes graph operations and information flow (Zhuge et al., 2024). QOBC covers single-round fixed graphs and pre-execution selection from a finite approved family. It does not control repeated debate or output-dependent graph expansion.

Structural efficiency. Graph search, pruning, and communication reduction can improve inference efficiency by modifying a system's configuration or interaction pattern (Zhang et al., 2025c;a; Wang et al., 2025b; Zhang et al., 2025b). Reshaping a workflow is distinct from allocating agent output budgets within a request-level spending limit during execution.

Budget-aware MAS design and allocation. BAMAS chooses models and collaboration topology under explicit budgets (Yang et al., 2026). FutureWeaver plans agent and collaboration-module calls using task state and lookahead (Jung et al., 2026). QOBC instead selects from an approved family of single-round graphs and controls per-call output limits, accounting for downstream input costs and revising unfinished allocations after actual usage is settled.

Online constrained execution. Replanning from observed consumption follows receding-horizon control (Rawlings et al., 2017). Separating predictive guidance from operational constraints is consistent with learning-augmented algorithms (Mitzenmacher & Vassilvitskii, 2022). QOBC uses standard constrained-optimization tools (Boyd & Vandenberghe, 2004) for provisional allocation, followed by integer feasibility repair and a separate dispatch check.

3 Problem Formulation

3.1 Quality Retention under a Request Budget

Consider a request x with an externally specified spending limit B. For a budget-control policy ϖ from the admissible class Π, let $Y_\varpi(x, B, \omega) \in [0, 1]$ be the final task score and $C_\varpi(x, B, \omega)$ its actual charge on execution realization ω. A refusal or incomplete workflow has score zero. The result-level objective is

$$\begin{aligned} \underset{\varpi \in \Pi}{\text{maximize}} \quad & J_\varpi(B) = \mathbb{E}_{x,\omega}[Y_\varpi(x, B, \omega)] \\ \text{subject to} \quad & C_\varpi(x, B, \omega) \le B \quad \text{for every admitted execution.} \end{aligned} \tag{1}$$

We seek a high $J_\varpi(B)$ throughout the intended budget range, so quality is retained as the limit tightens rather than cost being minimized without a specified limit. The area under a measured

budget–quality curve summarizes this behavior across levels. When a larger budget permits every lower-budget policy, the ideal value is nondecreasing in B; individual sampled outcomes need not be monotone.

3.2 Single-Round Execution Graphs and Budget–Quality Responses

Let $G = (V, E)$ be a single-round directed acyclic agent graph. Each vertex represents one agent instance with its own prompt and one model call; multiple agents may share the same instructions. The deployment supplies an ordered, finite family $\mathcal{F}(G)$ of admissible graphs and their message rules. Fixed topology is the special case $\mathcal{F}(G) = \{G\}$. Agents and connections remain fixed after a graph is selected. Agent i receives an integer output budget $b_i \in [\ell_i, u_i]$, and an enforced API cap gives $0 \le O_i \le b_i$.

For a fixed reference protocol, $Q_i(b_i)$ denotes the expected final-task response to changing only agent i's output budget. It is a conditional intervention response, not a separately graded probability that the agent's own message is correct. Section 4.1 learns these budget–quality curves. With frozen agent parameters Θ, a composition map places candidate plans on a common relative-quality scale,

$$U_\Theta(G', \mathbf{b}) = \mathcal{A}_\Theta\left(G', \{Q_i(b_i)\}_{i \in V(G')}\right). \tag{2}$$

This quality model guides planning toward the result-level objective in Eq. 1; its concrete form is given in Section 4.2 for optional execution-graph selection, evaluated in Section 5.2 and Appendix B.2 through held-out candidate-graph comparisons.

An output can incur both generation cost and later reading cost. Let a_i be the modeled known input length, including prompt, task, formatting, and an estimation margin. Let $\gamma_{ki} \ge 0$ give the input tokens at i induced by one output token from k; full forwarding under a shared tokenizer gives $\gamma_{ki} = 1$. For a candidate G', replacing unknown outputs by their limits yields

$$\widehat{I}_i(\mathbf{b}) = a_i + \sum_k \gamma_{ki} b_k, \qquad \widehat{C}_{G'}(\mathbf{b}) = C_0 + \sum_i \pi_i b_i, \tag{3}$$

where input and output prices $p_i^{\text{in}}, p_i^{\text{out}}$ define

$$C_0 = \sum_i p_i^{\text{in}} a_i, \qquad \pi_i = p_i^{\text{out}} + \sum_j p_j^{\text{in}} \gamma_{ij}. \tag{4}$$

All sums are over the candidate. Thus π_i prices a token at i together with its anticipated downstream readings. For request budget B, the continuous planning domain and joint quality-guided decision are

$$\mathcal{B}_{G'}(B) = \{\mathbf{b} : \boldsymbol{\ell} \le \mathbf{b} \le \mathbf{u},\ \widehat{C}_{G'}(\mathbf{b}) \le B\}, \tag{5}$$

$$\begin{aligned} \mathbf{b}^\star_{G'}(B) &\in \operatorname*{arg\,max}_{\mathbf{b} \in \mathcal{B}_{G'}(B)} U_\Theta(G', \mathbf{b}), \\ V_\Theta(G', B) &= U_\Theta\big(G', \mathbf{b}^\star_{G'}(B)\big), \\ G^\star &\in \operatorname*{arg\,max}_{\substack{G' \in \mathcal{F}(G) \\ \mathcal{B}_{G'}(B) \neq \emptyset}} V_\Theta(G', B). \end{aligned} \tag{6}$$

3.3 Hard Budget and Dispatch Affordability

The modeled constraint in Eq. 5 guides initial allocation; the hard spending limit concerns actual charged usage. At a settled batch boundary, let R_t be the available request budget and $\mathcal{J}_t$ the unfinished agents. A ready batch $A_t \subseteq \mathcal{J}_t$ contains calls whose parent messages are complete. Before dispatch, the adapter constructs those calls, bounds each actual input by $\overline{I}_i$, and applies an execution cap b_i^{exec} no greater than its planned cap $\widehat{b}_i$. The joint reservation condition is

$$\begin{aligned} L_t &= \sum_{i \in A_t} \left(p_i^{\text{in}} \overline{I}_i + p_i^{\text{out}} b_i^{\text{exec}}\right) \le R_t, \\ \ell_i &\le b_i^{\text{exec}} \le \widehat{b}_i. \end{aligned} \tag{7}$$

After all calls in the batch settle, actual charges $c_i = p_i^{\text{in}} I_i + p_i^{\text{out}} O_i$ give $R_{t+1} = R_t - \sum_{i \in A_t} c_i$. **Conditional budget guarantee.** With nonnegative prices, valid input bounds, enforced output caps, complete accounting, and unresolved charges kept reserved, atomic joint reservation ensures cumulative charged cost never exceeds B (Appendix A.2), regardless of forecast quality.

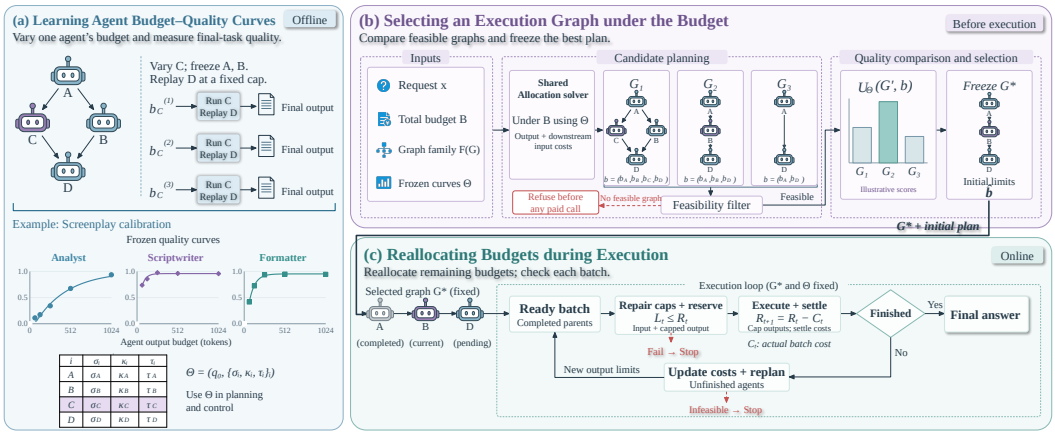


Figure 2: QOBC framework. (a) Learn and freeze agent budget–quality curves. (b) Compare feasible plans and freeze an execution graph. (c) Settle actual usage and reallocate remaining budgets with per-batch safety checks. The Screenplay inset uses calibration data; graph-comparison bars are schematic.

4 Quality-Aware Online Budget Control

Figure 2 summarizes QOBC's three stages. A shared allocator constructs candidate plans and updates unfinished-agent limits.

4.1 Learning Agent Budget–Quality Curves

Single-agent interventions estimate the conditional response in Section 3.2, modeled by a low-response region followed by diminishing returns, as

$$q_i(b_i) = q_0 + (1 - q_0)\left(1 - e^{-\kappa_i [b_i - \tau_i]_+}\right), \qquad Q_i(b_i) = \sigma_i q_i(b_i). \tag{8}$$

Here $[z]_+ = \max\{z, 0\}$, $q_0 \in (0, 1)$ is a shared floor, $\kappa_i > 0$ controls growth, $\tau_i \geq 0$ is a fitted shift, and $\sigma_i > 0$ sets the response scale. The shift describes the fitted curve; the execution floor ℓ_i is a separate deployment constraint. The frozen set is $\Theta = (q_0, \{\sigma_i, \kappa_i, \tau_i\}_i)$; Figure 3 illustrates the parameters and their distinct budget responses for three Screenplay content agents.

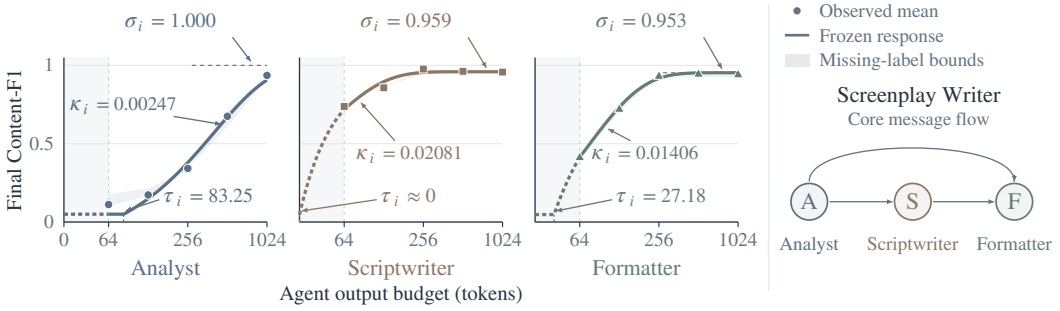


Figure 3: Screenplay content agents on 40 calibration inputs. Points average 36–40 valid Content-F1 labels; shading bounds missing labels by zero/one. Frozen fits mark shift τ_i, growth κ_i, and scale σ_i. Dashed segments below 64 tokens are unobserved extrapolations; the budget axis is linear to 64 and logarithmic thereafter. Right: message dependencies.

To estimate an agent's curve, calibration varies only that agent's output budget within a declared reference workflow. For task and upstream condition z, the prefix is executed once and its messages are frozen. Agent i is run at each grid budget b; affected downstream calls are then replayed under

common reference limits, while independent sibling outputs are held fixed. The fitting label $y_{i,z,b} \in [0,1]$ is the resulting final task score, so an upstream intervention includes its effect through the suffix. Multistart bounded least squares gives

$$(\widehat{\sigma}_i, \widehat{\kappa}_i, \widehat{\tau}_i) \in \operatorname*{arg\,min}_{\sigma,\kappa,\tau} \sum_{(z,b)\in\mathcal{D}_i} \left[\sigma q_i(b;\kappa,\tau) - y_{i,z,b}\right]^2. \tag{9}$$

Pooling z estimates the reference-condition mean with a common scoring rule; test labels never update parameters. Pipeline and Parallel use 512-token suffix limits, 40 shared calibration questions, and 50 disjoint test questions. Native reference settings, fitting splits, audits, and parameters appear in Appendices B.1 and C. We use $q_0 = .05$ for every fitted curve.

4.2 Selecting an Execution Graph under the Budget

To compare feasible execution plans, we instantiate Eq. 2 with a geometric mean of the learned budget–quality responses,

$$U_\Theta(G', \mathbf{b}) = \exp\left[\frac{1}{|V(G')|} \sum_{i\in V(G')} \log Q_i(b_i)\right]. \tag{10}$$

Normalizing by the participating-agent count avoids a product that decreases mechanically with graph size. The logarithmic composition gives greater marginal priority to improving a low response, while the effective token price captures its communication cost. For each affordable approved graph, the allocation mechanism below computes an initial budget plan. In a fixed-topology MAS this stage considers only the original graph; alternatives are used only when the deployment provides task-valid message rules for them. QOBC compares the plans using Eq. 10, resolves ties by the supplied order, and freezes the highest-scoring graph before execution. No candidate-specific structural prior is added. If no graph is feasible, the request is refused before a paid call.

4.3 Reallocating Budgets during Execution

After batch t, history h_t contains completed messages and metered usage. Completed messages enter $a_i(h_t)$; unfinished outputs are forecast from their limits, giving

$$\begin{aligned} \widehat{C}_t(\mathbf{b}) &= C_{0,t} + \sum_{i\in\mathcal{J}_t} \pi_{i,t} b_i, \\ C_{0,t} &= \sum_{i\in\mathcal{J}_t} p_i^{\text{in}} a_i(h_t), \\ \pi_{i,t} &= p_i^{\text{out}} + \sum_{j\in\mathcal{J}_t} p_j^{\text{in}} \gamma_{ij}. \end{aligned} \tag{11}$$

For a selected graph, its agent scales, completed-agent contributions, and the factor $1/|V(G')|$ are constant in the remaining allocation. Hence the surrogate subproblem is

$$\begin{aligned} \underset{\boldsymbol{\ell}\le\mathbf{b}_{\mathcal{J}_t}\le\mathbf{u}}{\text{maximize}} \quad & \sum_{i\in\mathcal{J}_t} \log q_i(b_i) \\ \text{subject to} \quad & \sum_{i\in\mathcal{J}_t} \pi_{i,t} b_i \le R_t - C_{0,t}. \end{aligned} \tag{12}$$

On a smooth growth branch, interior KKT stationarity gives

$$\begin{aligned} \frac{q_i'(b_i)}{q_i(b_i)} &= \lambda \pi_{i,t}, \\ b_i(\lambda) &= \tau_i + \frac{1}{\kappa_i} \log\left[(1-q_0)\left(1 + \frac{\kappa_i}{\lambda \pi_{i,t}}\right)\right]. \end{aligned} \tag{13}$$

Scalar bisection balances marginal log response per induced cost. Active-set handling enforces box bounds, followed by rounding down and feasibility repair. Affordable upper bounds are returned directly; unaffordable floors cause refusal. Appendix A.1 details this feasible integer approximation, branch handling, and concavity on the smooth growth branch.

Algorithm 1 QOBC execution-graph selection and online budget reallocation

Require: request x, ordered family $\mathcal{F}(G)$, budget B, frozen Θ

$h \leftarrow h_0(x)$, $R \leftarrow B$, $\mathcal{P} \leftarrow \emptyset$
for each $G' \in \mathcal{F}(G)$ **do**
  $\widehat{\mathbf{b}} \leftarrow \textsc{Solve}(G', R, h, \Theta)$
  if $\widehat{\mathbf{b}} \neq \perp$ **then**
    Add $(G', \widehat{\mathbf{b}}, U_\Theta(G', \widehat{\mathbf{b}}))$ to $\mathcal{P}$
  end if
end for
if $\mathcal{P} = \emptyset$ **then**
  return refusal without a paid call
end if
Select the highest-scoring graph and plan; freeze that graph
while unfinished agents remain **do**
  Form ready batch A using only completed-parent messages
  Project the current plan to modeled feasibility; check Eq. 7
  if either check fails **then**
    return refusal with the paid-prefix trace
  end if
  Commit the batch reserve and dispatch calls under their checked caps
  Settle actual usage of the complete batch; update h, R, and unfinished agents
  If agents remain, replan using Eq. 11; refuse if no plan exists
end while
return final answer and execution trace

Execution and budget safety. Adapters supply prompts, dependencies, candidate graphs, prices, and frozen parameters to one planner. Sequential agents form singleton batches; independent experts share completed-parent messages. The controller reduces caps toward floors as needed, jointly checks Eq. 7, dispatches, and settles usage. Replanning updates downstream costs with fixed graph and quality parameters; unaffordable continuation stops paid calls.

Using the learned curves, Algorithm 1 selects a feasible graph and initial limits, then alternates reservation, settlement, and budget reallocation. Quality predictions guide allocation; the separate accounting certificate ensures budget safety at each dispatch.

5 Experiments

We evaluate five MAS configurations (Table 1). Two controlled structures isolate allocation and graph-selection effects; three native applications test controller transfer with their original call order and prompts.

Pipeline and Parallel instantiate sequential handoff and parallel aggregation, respectively, reflecting collaboration patterns used in Chain of Agents (Zhang et al., 2024) and MoA (Wang et al., 2025a). Both are evaluated on MATH Level 3–5 (Hendrycks et al., 2021). Pipeline admits the full pipeline, Solver–Verifier, and Solver-only graphs; Parallel admits one to three experts between decomposition and aggregation. For native applications, we use the official MoA example on MATH, CrewAI Screenplay Writer (CrewAI, 2026) adapted to DialogSum (Chen et al., 2021), and the author's Corex Review (Sun et al., 2024) on GSM8K (Cobbe et al., 2021). Corex's five review calls form one forward traversal without revisiting an agent; its graph is not pruned. Native model backends are substituted as specified in Appendix C. Prompts, calibration data, and evaluation rules are fixed across methods within each configuration. Reduced Pipeline and Parallel graphs are explicitly supplied experimental alternatives, not claims that these MAS autonomously offer interchangeable routes. The three native applications expose only their original graph.

Budgets are 20/40/60/80/100% of each MAS's mean calibration-set cost, measured with predefined per-call limits but no request ceiling. Thus even 100% can constrain individual requests. Refusals

Table 1: MAS configurations used in our evaluation. Pipeline and Parallel are controlled structures with supplied graph alternatives; the three native applications retain singleton graph families.

MAS	Execution structure	Task	Model backend
Pipeline	Decompose, solve, verify	MATH	Doubao
Parallel	Decompose, experts, aggregate	MATH	Doubao
MoA	Four proposers, aggregator	MATH	Doubao + DeepSeek
Screenplay	Five sequential agents	DialogSum	DeepSeek
Corex Review	Five forward review calls	GSM8K	DeepSeek

and incomplete answers score zero. AUC is normalized over $[0.2, 1.0]$; paired 95% intervals use 10,000 question-level bootstrap resamples.

Four deployment-motivated controls separate allocation choices. Native-Stop preserves nominal limits and admits only affordable ready batches, motivated by quota checks in LLM gateways (Microsoft, 2026). Static Equal divides extra output expenditure once, modeling uniform quotas without role-utility estimates. Online Equal redistributes it after settlement, testing usage feedback alone. Historical Proportional uses reference output expenditures, motivated by usage-based provisioning such as Kubernetes VPA (Kubernetes Authors, n.d.). These are operational controls, not reproductions of the cited systems, and serve as distinct allocation baselines.

Static Quality freezes QOBC's initial quality-guided allocation but retains safety reductions. Its contrast with QOBC isolates online replanning; the optional graph-choice contrast is reserved for the ablation study (Appendix B.2).

5.1 Task Quality under Different Budget Constraints

Table 2 estimates $J_{\varpi}(B)$ in Eq. 1 at five budgets and summarizes the curve by AUC.

On Pipeline, QOBC exceeds all four deployment controls at 40–100% budgets, reaching 42% accuracy at 40% versus 2% for Online Equal. On Parallel, it exceeds these controls at every tier, retaining 40% accuracy at the tightest budget. Its AUC gains over Online Equal are 0.2200 ($[0.1550, 0.2900]$) and 0.2300 ($[0.1650, 0.2975]$), respectively, over the five budget levels.

On MoA, QOBC leads at every tier from 40% to 100%, with an AUC gain of 0.080 ($[0.053, 0.109]$) over Static Quality. On Screenplay, it exceeds Online Equal and Static Quality at 40–100%. Historical Proportional is stronger at 40–80% and in AUC, while QOBC leads at 100%, reaching 95.3% Content-F1 versus $[91.5, 92.5]$%. On Corex Review, QOBC leads at 40–100% and raises AUC from 0.5700 with Static Quality to 0.7388. Its paired gain is 0.1688 ($[0.1450, 0.1925]$). All Corex methods score zero at 20%. Native-Stop's zero completion on MoA and Corex reflects unaffordable nominal limits, not unconstrained reasoning performance of these systems.

5.2 Ablating Quality Guidance, Replanning, and Graph Choice

First, we hold each request's selected graph fixed. Online equal replaces quality guidance; static quality removes reallocation but retains safety reductions. These graph-matched policies run separately with identical accounting at 40/60/100% budgets, using AUC over $[.4, 1]$ rather than the main grid's $[.2, 1]$.

Table 3: Graph-matched accuracy AUC on budgets 40/60/100%, normalized over $[.4, 1]$. Each ablation uses the graph QOBC selected for the same request.

Allocation on the selected graph	Pipeline	Parallel
QOBC	**0.4800**	0.6167
Same graph, online equal	0.2333	0.5733
Same graph, static quality	0.4033	**0.6267**

Table 2: Task quality (%) at relative budgets and normalized AUC over $[.2, 1]$. Refusals and incomplete workflows score zero. Screenplay brackets bound 17 missing grades, not confidence intervals. Native applications retain their original graphs; graph choice is ablated in Appendix Table 5.

Method	20%	40%	60%	80%	100%	AUC
Pipeline (accuracy, $n = 50$)						
Native-Stop	0.0	0.0	0.0	0.0	10.0	0.0125
Static Equal	0.0	0.0	4.0	16.0	34.0	0.0925
Online Equal	**2.0**	2.0	10.0	34.0	48.0	0.1775
Historical Proportional	0.0	0.0	12.0	24.0	44.0	0.1450
QOBC	0.0	**42.0**	**38.0**	**46.0**	**66.0**	**0.3975**
Parallel (accuracy, $n = 50$)						
Native-Stop	0.0	0.0	0.0	0.0	22.0	0.0275
Static Equal	0.0	0.0	14.0	26.0	52.0	0.1650
Online Equal	2.0	22.0	42.0	52.0	64.0	0.3725
Historical Proportional	2.0	22.0	32.0	44.0	56.0	0.3175
QOBC	**40.0**	**56.0**	**58.0**	**72.0**	**70.0**	**0.6025**
MoA (accuracy, $n = 100$)						
Native-Stop	0.0	0.0	0.0	0.0	0.0	0.0000
Static Equal	0.0	0.0	0.0	7.0	16.0	0.0375
Online Equal	0.0	0.0	5.0	13.0	23.0	0.0738
Historical Proportional	0.0	4.0	6.0	14.0	26.0	0.0925
Static Quality	0.0	24.0	67.0	81.0	86.0	0.5375
QOBC	0.0	**37.0**	**74.0**	**92.0**	**88.0**	**0.6175**
Screenplay (Content-F1, $n = 100$)						
Native-Stop	0.0	0.0	0.0	3.2	[52.7, 53.7]	[0.0738, 0.0751]
Static Equal	0.0	2.2	[21.4, 22.4]	42.7	61.8	[0.2434, 0.2459]
Online Equal	0.0	10.4	[23.6, 26.6]	[43.8, 45.8]	68.4	[0.2799, 0.2924]
Historical Proportional	0.0	**[16.4, 18.4]**	**60.1**	**[82.2, 84.2]**	[91.5, 92.5]	**[0.5111, 0.5223]**
Static Quality	0.0	4.8	[39.9, 40.9]	[71.8, 72.8]	[87.7, 88.7]	[0.4009, 0.4071]
QOBC	0.0	14.2	[47.3, 48.3]	[78.3, 79.3]	**95.3**	[0.4687, 0.4737]
Corex Review (GSM8K accuracy, $n = 100$)						
Native-Stop	0.0	0.0	0.0	0.0	0.0	0.0000
Static Equal	0.0	0.0	0.0	9.0	35.0	0.0663
Online Equal	0.0	9.0	35.0	66.0	81.0	0.3763
Historical Proportional	0.0	8.0	43.0	65.0	84.0	0.3950
Static Quality	0.0	3.0	86.0	94.0	90.0	0.5700
QOBC	0.0	**62.0**	**92.0**	**95.0**	**93.0**	**0.7388**

On Pipeline, paired AUC gains are 0.2467 ($[0.1700, 0.3300]$) over online equal and 0.0767 ($[0.0167, 0.1400]$) over static quality. Parallel's corresponding intervals, $[-0.0100, 0.0967]$ and $[-0.0567, 0.0333]$, include zero. Pipeline supports both components; Parallel's limited redistribution opportunity is examined through execution traces in Section 5.3.

Second, we isolate optional graph selection while keeping QOBC's allocator, online replanning, and budget checks unchanged. Fixed-graph QOBC uses the original full graph; full QOBC can choose from the supplied alternatives. This comparison uses the five-tier main protocol rather than the graph-matched three-tier protocol above, which isolates component effects.

Graph choice leaves Pipeline AUC essentially unchanged (0.3975 versus 0.4000; paired difference -0.0025, 95% interval $[-0.0300, 0.0250]$). On Parallel, it raises AUC from 0.5275 to 0.6025 (difference $+0.0750$, $[0.0100, 0.1350]$), despite the fixed graph winning at 40% and 60%. Graph choice therefore requires meaningful approved alternatives and is not needed for fixed-topology deployment. Figure 4 illustrates Pipeline's selected variants; full tiers and selection diagnostics appear in Appendix B.2, including per-budget counts and ranking outcomes.

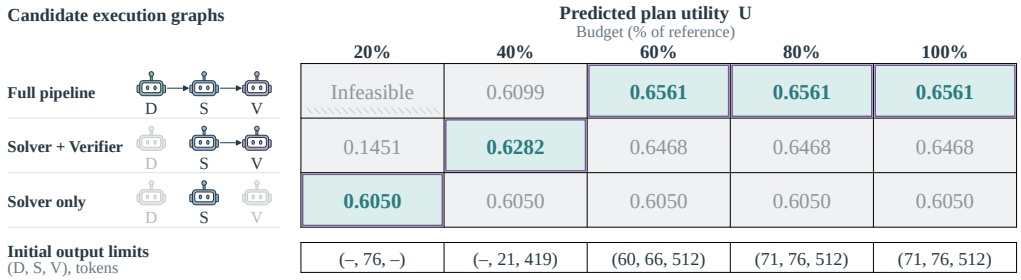


Figure 4: One Pipeline input under five budgets and supplied graph alternatives. Color marks selected plans; gray marks alternatives. Scores are predicted U_Θ, not observed accuracy. Initial limits list Decomposer, Solver and Verifier (dash: absent); the selected graph remains fixed during execution.

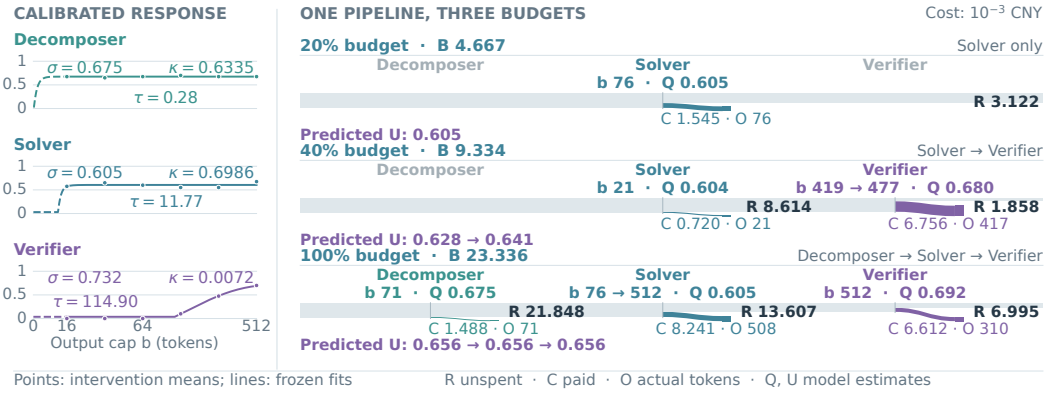


Figure 5: Controlled Pipeline with supplied graph alternatives. Left: 40-input intervention means and fits (below 16 tokens: extrapolation). Right: one input across budgets; R is unspent budget, C paid cost (row-normalized), b the output limit, O actual output, and Q, U model estimates, not scores.

5.3 Execution Dynamics and Budget Safety

Figure 5 illustrates allocation and settlement, not an aggregate graph-choice gain. At 40%, settlement raises Verifier's limit from 419 to 477 and U from 0.628 to 0.641; at 100%, Solver's limit rises from 76 to 512 while saturated U stays at 0.656. Mean Pipeline gains are 58.8 tokens for Verifier at 40% (46 calls) and 427.3 for Solver at 100%; Parallel's terminal gain is at most 0.6. All 9,000 native and 4,200 structural conditions respect the budget; structural input/output checks pass, and QOBC completes all 500 structural main requests. Median Pipeline/Parallel planning times are 24.1/33.2 ms without API latency. Appendices B.3 and A.2 detail the accounting.

6 Discussion and Conclusion

QOBC combines learned budget–quality curves and online reallocation under hard request budgets, with optional graph choice for valid alternatives. It improves on online equal allocation in both controlled structures and static quality allocation in all three native applications. Graph choice is neutral on Pipeline and favorable overall on Parallel, but depends on task-valid alternatives and imperfect cross-graph quality estimates. Historical allocation has higher overall Screenplay AUC, although QOBC leads at 100% budget. MoA answer-form review and 17 bounded Screenplay grades warrant human validation. Safety requires valid input bounds, enforced caps, and complete billing records; it does not ensure completion of every admitted workflow.

AI USE STATEMENT

Generative AI assisted literature discovery, research ideation and experimental design, controller and experimental-code implementation, figure design, and manuscript drafting and revision. Model-based grading is disclosed in Appendix C. The authors reviewed the AI-assisted work and remain responsible for the final content and results.

REPRODUCIBILITY STATEMENT

Appendices B and C specify splits, pricing, calibration, and accounting. Artifacts retain source hashes and per-call usage.

REFERENCES

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. URL https://web.stanford.edu/~boyd/cvxbook/.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 5062–5074, 2021. doi: 10.18653/v1/2021.findings-acl.449. URL https://aclanthology.org/2021.findings-acl.449/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

CrewAI. Screenplay writer: Official CrewAI example. Source code, crewAI-examples repository, 2026. URL https://github.com/crewAIInc/crewAI-examples/tree/da94a91e691e1cf5b3151416bb15b5b62729bea8/crews/screenplay_writer. Accessed September 24, 2026; revision da94a91.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*, 2024.

Dongwon Jung, Peng Shi, Muhao Chen, and Yi Zhang. FutureWeaver: Planning test-time compute for multi-agent systems with modular collaboration. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2512.11213v2. Accepted paper; preprint version 2, June 2026.

Kubernetes Authors. Vertical pod autoscaler: Components. Official Kubernetes autoscaler documentation, n.d. URL https://github.com/kubernetes/autoscaler/blob/master/vertical-pod-autoscaler/docs/components.md. Accessed September 26, 2026.

Microsoft. Limit large language model API token usage. Microsoft Learn, Azure API Management documentation, 2026. URL https://learn.microsoft.com/en-us/azure/api-management/llm-token-limit-policy. Updated May 6, 2026. Accessed September 26, 2026.

Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *Communications of the ACM*, 65(7):33–35, 2022. doi: 10.1145/3528087.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186, 2024. doi: 10.18653/v1/2024.acl-long.810.

James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2nd edition, 2017.

Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration. In *Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=7BCmIWVT0V.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5434be94e82c54327bb9dcaf7fca52b6-Abstract-Conference.html.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. AgentDropout: Dynamic agent elimination for token-efficient and high-performance LLM-based multi-agent collaboration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 24013–24035, 2025b. doi: 10.18653/v1/2025.acl-long.1170.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *Conference on Language Modeling (COLM)*, 2024.

Liming Yang, Junyu Luo, Xuanzhe Liu, Yiling Lou, and Zhenpeng Chen. BAMAS: Structuring budget-aware multi-agent systems. In *Proceedings of the 40th Annual AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 29802–29810, 2026. doi: 10.1609/aaai.v40i35.40226.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. Multi-agent architecture search via agentic supernet. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 75834–75852, 2025a.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. In *International Conference on Learning Representations (ICLR)*, 2025b.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *International Conference on Learning Representations (ICLR)*, 2025c.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arik. Chain of agents: Large language models collaborating on long-context tasks. In *Advances in Neural Information Processing Systems*, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/ee71a4b14ec26710b39ee6be113d7750-Abstract-Conference.html.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62743–62767, 2024.

A Essential Derivations and Safety Guarantee

A.1 Response Composition and Numerical Allocation

Each fitted $Q_i(b)$ estimates expected final-task score under a fixed reference workflow, prefix and sibling messages, suffix limits, and scorer. Affected descendants are rerun after the target intervention. The geometric composition is a ranking surrogate, not a joint success probability or an independence assumption. No candidate-specific structural prior is used.

For a fixed graph, agent scales, completed-agent contributions, and the normalization factor are constant in the remaining optimization. With $R_{\text{free}} = R_t - C_{0,t}$, the smooth-branch Lagrangian is

$$\mathcal{L}(\mathbf{b}, \lambda) = \sum_{i \in \mathcal{J}_t} \log q_i(b_i) - \lambda \left(\sum_{i \in \mathcal{J}_t} \pi_{i,t} b_i - R_{\text{free}} \right).$$

For $b_i > \tau_i$, let $r_i = (1 - q_0)e^{-\kappa_i(b_i - \tau_i)}$. Then $q_i = 1 - r_i$ and

$$\begin{aligned} \frac{\kappa_i r_i}{1 - r_i} &= \lambda \pi_{i,t}, \\ r_i &= \frac{\lambda \pi_{i,t}}{\kappa_i + \lambda \pi_{i,t}}, \\ \frac{d^2 \log q_i}{db_i^2} &= -\frac{\kappa_i^2 r_i}{(1 - r_i)^2} < 0. \end{aligned}$$

This yields Eq. 13. The continuous objective is concave when all permitted intervals lie on the growth branch. At τ_i, the derivative jumps upward from zero, so intervals crossing the shift need not be concave. The branch-based routine is therefore a feasible integer approximation, not a general global optimizer across transitions.

For a trial multiplier, the allocator evaluates active growth branches, clamps allocations to their bounds, and bisects with active-set handling of saturated agents. Inactive branches use their floors. It rounds down and repairs modeled feasibility; affordable upper bounds are returned directly and unaffordable floors cause refusal. With m unfinished agents and K bisection iterations, repeated active-set passes have conservative cost $O(m^2 K)$ per candidate. For integer plan $\widehat{\mathbf{b}}$, runtime projection searches reductions $b_i(\alpha) = \ell_i + \lfloor \alpha(\widehat{b}_i - \ell_i) \rfloor$, $\alpha \in [0, 1]$. Static controls cannot increase their original limits.

Proposition 1. *For fixed response parameters and candidate graphs, the ideal value in Eq. 6 is nondecreasing in budget wherever a feasible candidate exists.*

Proof. If $B_1 \le B_2$, each candidate's feasible set satisfies $\mathcal{B}_{G'}(B_1) \subseteq \mathcal{B}_{G'}(B_2)$. Maximization over an enlarged feasible set cannot lower its value; maximization over the candidate family preserves the ordering. □

This concerns the ideal surrogate, not monotonicity of sampled task scores or approximate solver outputs.

A.2 Conditional Spending Guarantee

Theorem 1. *Assume known nonnegative prices, enforced output limits, valid input bounds for every dispatched call, and complete billable-usage accounting. If ready batches are atomically reserved under Eq. 7 and unresolved charges remain reserved, cumulative charged expenditure cannot exceed B.*

Proof. Let S be settled cost, P outstanding reservations, and $R = B - S - P$ available funds. Initially $S = P = 0$. A batch reserve $L \le R$ moves from R to P before dispatch. For an attempt reserved at l, valid input bounds and enforced output limits imply charge $c \le l$. Settlement adds c to S, removes l from P, and returns $l - c$ to R. Thus $S + P + R = B$ and all terms remain nonnegative. Concurrent calls cannot reuse reserved funds; unresolved attempts retain their

reservations and retries require new ones. Refusal dispatches no call. Induction over dispatch and settlement gives $S \leq B$, including at termination. □

Quality predictions and future-input forecasts guide planning but are not assumptions of this spending proof. Safety does not imply completion: new messages can render an unfinished suffix unaffordable. Input-bound or output-limit breaches must be reported, never hidden by clipping usage. An invoice-level guarantee requires the same invariant for all billable attempts, not only successful logical trajectories.

B Essential Experimental Protocol

B.1 Controlled Workflows and Calibration

Pipeline's ordered candidates are full_pipeline, solve_verify, solve_only. Parallel uses three_experts, two_experts, one_expert; reduced variants retain the first experts in the declared order together with Decomposer and Aggregator. This order breaks proxy ties. Ready experts share a completed parent snapshot and settle as a batch. Candidate families and message rules are supplied deployment inputs, not arbitrary agent deletions.

Both structures use 40 shared MATH Level 3–5 calibration questions and 50 disjoint evaluation questions, with selection seeds 2026092417 and 2026092418. Generation uses `doubao-seed-2-1-turbo-260628`, temperature zero and disabled thinking. Logical input/output prices are 3/15 CNY per million tokens. Preflight adds a 192-token provider margin and 24 tokens per parent edge. Forty nominal references with 512-token per-agent limits and no request ceiling define mean costs 0.023336175 CNY and 0.04001625 CNY. Each method receives $B_r = r\bar{C}_G$ for the same $r \in \{.2, .4, .6, .8, 1\}$.

Calibration freezes prefixes and independent siblings, varies one agent's output limit over $\{16, 32, 64, 128, 256, 512\}$, and reruns affected descendants at the reference limits. All 80 references and 1,920 interventions completed. The first 32 questions fit the curves and eight validate them before refitting on all 40. Fourteen fixed starts combine seven shifts and two growth rates; the lowest squared-error fit is selected within $\sigma_i \in [.01, 1]$, $\kappa_i \in [10^{-5}, 1]$, $\tau_i \in [0, 512]$, with $q_0 = .05$. Execution floors remain separate from fitted shifts. Table 4 records rounded parameters; frozen configurations retain full precision, input IDs, source hashes, and generation settings.

The structural reference audit corrects 22 balanced-boxed answers. Sixty-nine intervention comparisons on two questions required a method-blind answer-form rule: `Empty`/`Empty set` match `empty`, and `23000` matches 23,000 dollars. All other raw outputs and original scores are retained; no unsupported forms remain unresolved in this batch.

Held-out fit check. Each agent has 48 validation observations from eight questions and six limits. Against a constant fitting-label mean, curve MSE changes from 0.153 to 0.085 for Pipeline's Verifier and from 0.178 to 0.075 for Parallel's Aggregator. Upstream errors remain close to the constant reference. This assesses individual responses; composed graph scores are evaluated separately in Appendix B.2.

B.2 Controls, Statistical Units, and Graph Metrics

Native-Stop keeps nominal limits and checks the next ready batch only. Static Equal reserves modeled minimum cost and divides extra output expenditure equally once, accounting for induced inputs and upper bounds; Online Equal repeats this after settlement. Historical Proportional uses reference mean output expenditures, lower-bounded by minimum-output expenditure, and updates online. Graph-matched ablations lock QOBC's chosen graph; static quality also locks its initial allocation except for safety reductions. All retain common prices, bounds and dispatch checks. Fixed-graph QOBC is a separate graph-choice ablation: it retains the original full graph with the same allocator and online updates as QOBC. Only the controlled workflows have more than one approved candidate. They execute separately, so paired effects compare policies, not a deterministic single-step counterfactual.

Table 4: Frozen intervention-response parameters. Labels are final-task scores under reference conditions. Native values are rounded here; full precision remains in frozen configurations. Execution floors are separate from fitted shifts.

Workflow	Agent	σ_i	κ_i	τ_i
Pipeline	Decomposer	0.6750	0.63347	0.28
Pipeline	Solver	0.6050	0.69860	11.77
Pipeline	Verifier	0.7319	0.00717	114.90
Parallel	Decomposer	0.7550	0.30924	0.00
Parallel	Solver1	0.7054	0.19261	0.00
Parallel	Solver2	0.7051	0.29887	0.00
Parallel	Solver3	0.7472	0.14020	0.00
Parallel	Aggregator	0.7730	0.01303	118.41
MoA	Doubao 1	0.9375	0.99655	0.01
MoA	DeepSeek 1	0.9750	0.23269	11.84
MoA	Doubao 2	0.9583	0.19979	41.22
MoA	DeepSeek 2	0.9625	0.65049	0.00
MoA	aggregator	0.9714	0.01375	837.82
Screenplay	analyst	1.0000	0.00247	83.25
Screenplay	scriptwriter	0.9591	0.02081	0.00
Screenplay	formatter	0.9530	0.01406	27.18

For mean score $a(r)$ on budget grid $r_1 < \cdots < r_K$, normalized AUC is

$$\mathrm{AUC} = \frac{1}{r_K - r_1} \sum_{k=1}^{K-1} (r_{k+1} - r_k) \frac{a(r_k) + a(r_{k+1})}{2}.$$

Main comparisons use $[.2, 1]$ and matched ablations $[.4, 1]$. Bootstrap draws resample whole questions with all paired methods, budgets and candidates. Intervals condition on frozen calibration and do not include refitting uncertainty. Refusals and incomplete answers score zero; infrastructure missingness is reported separately. Repeated conditions and cached calls do not increase the independent sample count.

For question–budget group z, let $\mathcal{F}_z^+$ be initially feasible candidates, $\widehat{G}_z$ the proxy choice before observing outcomes, and $y_z(G')$ the measured score. Define

$$y_z^{\mathrm{best}} = \max_{G' \in \mathcal{F}_z^+} y_z(G'),$$
$$\mathrm{regret}_z = y_z^{\mathrm{best}} - y_z(\widehat{G}_z),$$
$$\mathrm{hit}_z = \mathbf{1}\{y_z(\widehat{G}_z) = y_z^{\mathrm{best}}\}.$$

Best-score ties count as hits. No-feasible-candidate groups remain in overall denominators with score and regret zero; feasible-group hits condition on at least one candidate. Pair agreement counts only pairs with unequal proxy scores and unequal observed outcomes. All 600 candidate executions completed, giving 100 feasible groups per workflow. Pipeline and Parallel have 53 and 25 all-zero groups, 93 and 81 best-score ties, and 21 and 24 groups with informative pairs. These denominators accompany the main-text regret and agreement results. The observed best is a hindsight summary, not an unbiased optimal expected-value policy.

Table 5: Optional graph-choice ablation on the same 50 questions per controlled workflow. Fixed-graph QOBC uses the original full graph; full QOBC may select a supplied alternative. Accuracy is in percent; AUC is normalized over $[.2, 1]$.

Graph policy	20%	40%	60%	80%	100%	AUC
Pipeline (accuracy, $n = 50$)						
Fixed graph	**2.0**	**42.0**	**38.0**	**46.0**	**66.0**	**0.4000**
Graph choice	0.0	**42.0**	**38.0**	**46.0**	**66.0**	0.3975
Parallel (accuracy, $n = 50$)						
Fixed graph	4.0	**60.0**	**64.0**	56.0	58.0	0.5275
Graph choice	**40.0**	56.0	58.0	**72.0**	**70.0**	**0.6025**

Graph-choice diagnostics. Figure 4 illustrates the supplied Pipeline variants, not a native Pipeline capability. Across the main evaluation, Pipeline selects Solver alone at 20%, Solver–Verifier for 46/50 requests at 40%, and the full graph from 60%; Parallel retains one expert throughout. For a separate ranking probe, every approved graph runs on 50 held-out questions at 40% and 80% with initial allocation and no replanning. Selected/observed-best mean scores are 0.41/0.47 on Pipeline and 0.66/0.75 on Parallel; mean regrets are 0.06 ($[0.01, 0.12]$) and 0.09 ($[0.04, 0.15]$). Top-1 hits of 94%/91% include 93/81 tied groups out of 100, while informative-pair agreement is 20/42 and 33/48. Selection often reaches a tied best outcome, but these results do not establish universal ranking accuracy. Appendix B.1 assesses individual fits.

B.3 Accounting and Trace Selection

Cached reuse requires the same complete provider request, including model, messages, temperature and output limit. Usage is charged within every logical trajectory, independently of cache-source selection. The 4,200 structural conditions comprise 3,000 main comparisons, 600 matched ablations and 600 graph probes. Their answer-form audit resolves 94 labels. The main grid has 2,192 completions, 429 pre-call refusals and 379 paid-prefix stops; mean utilization including refusals is 0.643. No request-budget, input-bound or output-limit violation is observed. Planning excludes API latency; median/95th-percentile times are 24.1/60.9 ms for Pipeline and 33.2/55.4 ms for Parallel.

Logical policy costs are not the research invoice. The research ledger records 8,282 settled calls costing 37.762131 CNY and 33 failed attempts with unresolved provider billing, retaining 0.198885 CNY of reserve under a separate 150-CNY ceiling. They are not silently treated as free retries. Figure 5 retains the prior Pipeline case chosen by the lowest question-text hash among eligible multi-stage paired requests with different terminal limits, without using quality labels or the sign of the limit change. Its 40% trace is joined by the same input at the protocol endpoints, 20% and 100%; all five budget traces are retained in the artifact. Reserved funds are not paid charges. Responses use the frozen full-precision parameters. At each planning step, U_Θ is recomputed from logged safe limits for unfinished agents and dispatched limits for completed agents, not their actual output lengths. Initial values match the logged candidate utilities. Subsequent values are reconstructed model estimates, not observed quality or newly fitted predictions. These cases illustrate execution, separately from aggregate effect estimates.

C Essential Native-Workflow Settings and Scoring

C.1 Workflow Preservation and Calibration

MoA retains the official `moa.py` revision `050d087`, four concurrent proposers, and the subsequent aggregation prompt and message order. Two proposer slots use `doubao-seed-2-1-turbo-260628`, two use `deepseek-flash`, and the aggregator uses Doubao. These are four instances of two models, not four model families. Proposer temperature is 0.7 and aggregator temperature is unspecified as in the source. Nominal limits are 512 per proposer and the backend-default 4,096 for the aggregator. This substitutes backends while preserving the source workflow; it does not reproduce the original paper's benchmark.

Screenplay uses revision `da94a91` with CrewAI 1.15.6. Compatibility repairs bind dialogue/script placeholders, adapt deprecated interfaces, and accept a successful empty filter response without an extra repair call. Instructions, cumulative contexts and the source's print-only STOP behavior are retained. All five agents use `deepseek-flash`, temperature zero, and disabled thinking. Nominal limits for filter, analyst, scriptwriter, formatter and scorer are 32/2,048/512/512/64. Content-agent maxima round reference usage upward to powers of two; auxiliary limits are common deployment settings. Only the native scorer's first line is consumed, and that score is not used as the evaluation label.

Corex Review uses the author's revision `7fb3730` and its five-call forward chain: one seeded initial agent, followed by four sequential reviewers. The task and review prompts, reviewer order, temperature zero, and final-answer extraction are retained. The provider is replaced by `deepseek-flash`, with a common 4,096-token nominal output limit. Generated-code execution is blocked without an isolated executor; the completed GSM8K evaluation does not enter this branch. This evaluates the Review entry point, not Corex's Discuss or Retrieve modes, and does not reproduce the original paper's model results.

All three applications use the same controller with singleton candidate families. Adapters supply messages, dependencies, prices and fitted responses; auxiliary Screenplay calls have fixed, fully charged limits. Forty reference runs give mean costs 0.03761635 CNY for MoA, 0.022870 CNY for Screenplay, and 0.01686325 CNY for Corex. Fixed input/output prices per million tokens are 3/15 CNY for Doubao and 2/8 CNY for DeepSeek. Each 100-input test cohort is disjoint from calibration, development and earlier pilots. Split/execution seeds are 2026092403/2026092406 for MoA and 20260924 for Screenplay; exact IDs and source hashes remain frozen.

MoA and Screenplay calibration fits on 32 of 40 inputs, checks eight, then refits all available calibration labels. MoA proposer grids are 64/128/256/512 and aggregator grids 64/256/1,024/4,096. Screenplay core-agent grids are 64/128/256/512/1,024, with 4,096-token reference limits. Prefixes and independent siblings stay fixed while affected descendants are rerun. MoA has 839 valid labels out of 840 conditions; Screenplay has 625/640, with 14 invalid grades and one generation failure. Missing labels are not assigned zero. Full-precision fits and configurations are retained. Corex uses 40 calibration questions and 800 role-cap intervention conditions. After finding that the pilot's numeric extractor missed marked final answers, these calibration labels were rescored and the response curves refitted offline. No calibration API calls were repeated. Its independent confirmation uses 100 previously unused entries from the official GSM8K test file, disjoint from two engineering checks, 40 calibration questions, and 20 pilot questions. It is an internal test partition, not a full-benchmark replication.

C.2 Scoring and Missing-Label Bounds

MoA uses balanced-brace reference parsing and boxed-answer extraction, falling back to an answer-only extractor that sees neither the question nor the reference. Among 1,682 complete outputs, 102 answer-form comparisons across 25 questions were flagged. Post-test, method-agnostic assistant review classified 92 as equivalent and ten as incorrect without changing generations or fitted parameters. This is not independent human adjudication. Strict scoring gives QOBC/Static Quality/Historical Proportional AUCs 0.545/0.469/0.069. Treating all flagged comparisons as unknown gives $[0.545, 0.629]$/$[0.469, 0.544]$/$[0.069, 0.095]$. Also treating 43 positive, unboxed, length-stopped outputs as unknown leaves the QOBC–Static Quality difference in $[0.054, 0.124]$.

Screenplay evaluates content preservation, not dramatic quality. Before test generation, 100 DialogSum source-fact lists were constructed; three nonliteral excerpts were corrected against the dialogues, yielding 1,521 facts with original responses preserved. An external Doubao grader sees source, facts and script but not method or budget. Recall measures fact coverage, precision measures supported script claims with correct speaker attribution, and Content-F1 is their harmonic mean. This is not a human-validated gold standard. Of 2,059 completed scripts, 2,042 pass grade validation; 14 unsupported evidence quotes, two invalid annotations and one rejected call leave 17 missing grades. All 941 budget-stopped requests score zero. Brackets are worst/best missing-label bounds, not confidence intervals. Table 2 reports all methods and budget tiers; Table 6 additionally reports completion counts.

Table 6: MoA and Screenplay transfer on 100 new inputs each. AUC is normalized over $[.2, 1]$; completion counts input–budget conditions. MoA uses answer-form review; Screenplay brackets bound missing grades. Corex Review appears in Table 2.

	MoA (accuracy)		Screenplay (Content-F1)	
Method	AUC	Completed	AUC	Completed
Native-Stop	0.000	0/500	[0.074, 0.075]	59/500
Static Equal	0.038	253/500	[0.243, 0.246]	**400/500**
Online Equal	0.074	260/500	[0.280, 0.292]	**400/500**
Historical Proportional	0.093	389/500	**[0.511, 0.522]**	**400/500**
Static Quality	0.538	**390/500**	[0.401, 0.407]	**400/500**
QOBC	**0.618**	**390/500**	[0.469, 0.474]	**400/500**

Corex uses exact final-numeric matching on GSM8K, accepting a number only after the last explicit answer cue and allowing Markdown, TeX, or currency formatting. Incomplete and refused requests score zero. This rule was frozen before the independent 100-question comparison; no LLM judge is used. All 3,000 question–method–budget cells have a recorded result. An offline audit recomputes every label from its output, checks each request budget, and finds no violation. The paired 95% AUC interval uses 10,000 question-level bootstrap resamples. Corex has no approved subgraph choice, so this comparison evaluates allocation and replanning rather than graph selection.

MoA and Screenplay each complete their 3,000-condition generation grids, with 1,682 MoA and 2,059 Screenplay completions and the remainder budget-stopped. No test-generation infrastructure failure, unsettled charge or budget violation is recorded. Logical costs are 31.766242/24.392180 CNY, excluding calibration and grading. Static Quality's initial limits match QOBC in all 500 input–budget pairs per application; both retain safety projection.