# WWTP OPERATOR CHALLENGE

*Can AI Run a Wastewater Treatment Plant?*

A Kaggle Benchmark Testing 22 AI Models
on 30 Days of Real-World Plant Operations

Based on: Ceyhan Wastewater Treatment Plant (Adana, Turkey)
Capacity: 50,000 m³/day | Simulation: Monod Kinetics & Mass Balance

**February 2026**
Kaggle Benchmarks Team

# 1. Executive Summary

**We put 22 AI models in charge of a real wastewater treatment plant for 30 days.** The results were humbling: the best model scored only 64 out of 100, and the average score was just 39. No model achieved what a competent human operator would consider a passing grade.

The WWTP Operator Challenge is a Kaggle benchmark that tests whether large language models (LLMs) can make sound operational decisions in a complex, dynamic engineering environment. Unlike typical AI benchmarks that test knowledge recall or code generation, this benchmark requires models to manage a living biological system where every decision has cascading consequences over time.

Each model operated the Ceyhan Wastewater Treatment Plant in Adana, Turkey (50,000 m³/day capacity) through a 30-day simulation that included four major crisis events: rain flooding, an industrial shock load, a cold weather spell, and a mechanical blower failure. Models had to make 5 daily decisions — controlling aeration, sludge recycling, sludge wasting, chemical dosing, and emergency bypass — while keeping effluent discharge within Turkish legal limits.

**Key takeaway:** AI models demonstrated textbook knowledge of wastewater treatment but failed at the practical, experience-driven decision-making that real operators excel at. When multiple crises hit simultaneously, most models panicked, made counterproductive decisions, or hallucinated solutions that didn't exist within the system.

| 22 | 64/100 | 39/100 | 26/100 |
|---|---|---|---|
| Models Tested | Best Score | Average Score | Worst Score |

# 2. How the Benchmark Works

## 2.1 The Simulation

The benchmark simulates a real activated sludge wastewater treatment plant using Monod kinetics and mass balance equations. The simulation engine tracks bacterial population dynamics (MLSS), dissolved oxygen levels, sludge age (SRT), sludge settleability (SVI), and calculates effluent quality for five regulated parameters: COD, BOD, TSS, Total Nitrogen, and Total Phosphorus.

Each day, the model receives a detailed status report showing influent water quality, current plant conditions, yesterday's effluent results, and energy consumption. The model must respond with exactly 5 operational decisions in JSON format. Decisions from one day directly affect the next day's plant state — creating a 30-step sequential decision chain where early mistakes compound over time.

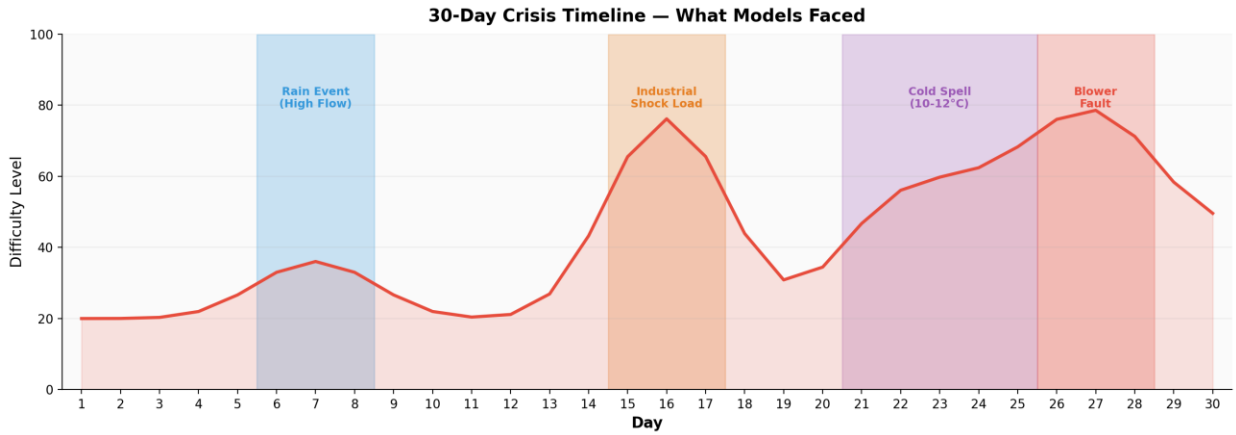
## 2.2 Scoring System

The scoring system evaluates four dimensions of operational competence, totaling 100 points:

| Category | Points | How It's Measured |
|---|---|---|
| Discharge Compliance | 40 | Each parameter violation costs 2 points; each bypass day costs 5 points |
| Energy Efficiency | 20 | Based on average kWh/m³: <0.40 = 20p, 0.40-0.50 = 16p, 0.50-0.55 = 12p, >0.65 = 4p |
| Biological Stability | 20 | Penalties for MLSS outside 1500-6000, SRT outside 4-30 days, SVI > 200 |
| Crisis Management | 20 | Extra penalties for bypass during shock load (Days 15-17) and violations during blower fault (Days 26-28) |

## 2.3 The 30-Day Crisis Timeline

The scenario was designed to test progressive escalation. The first week is relatively calm, giving models time to establish a stable baseline. Then, four crisis events hit with increasing severity:




**Days 1-5:** Calm baseline period with normal influent conditions (COD ~300, BOD ~150 mg/L, 21°C).

**Days 6-8:** Rain event increases flow to 55,000+ m³/day with diluted but high-volume influent.

**Days 15-17:** Industrial shock load — the hardest test. COD spikes to 800 mg/L, BOD to 400, pH drops to 6.2. This simulates an illegal industrial discharge that real plants occasionally face.

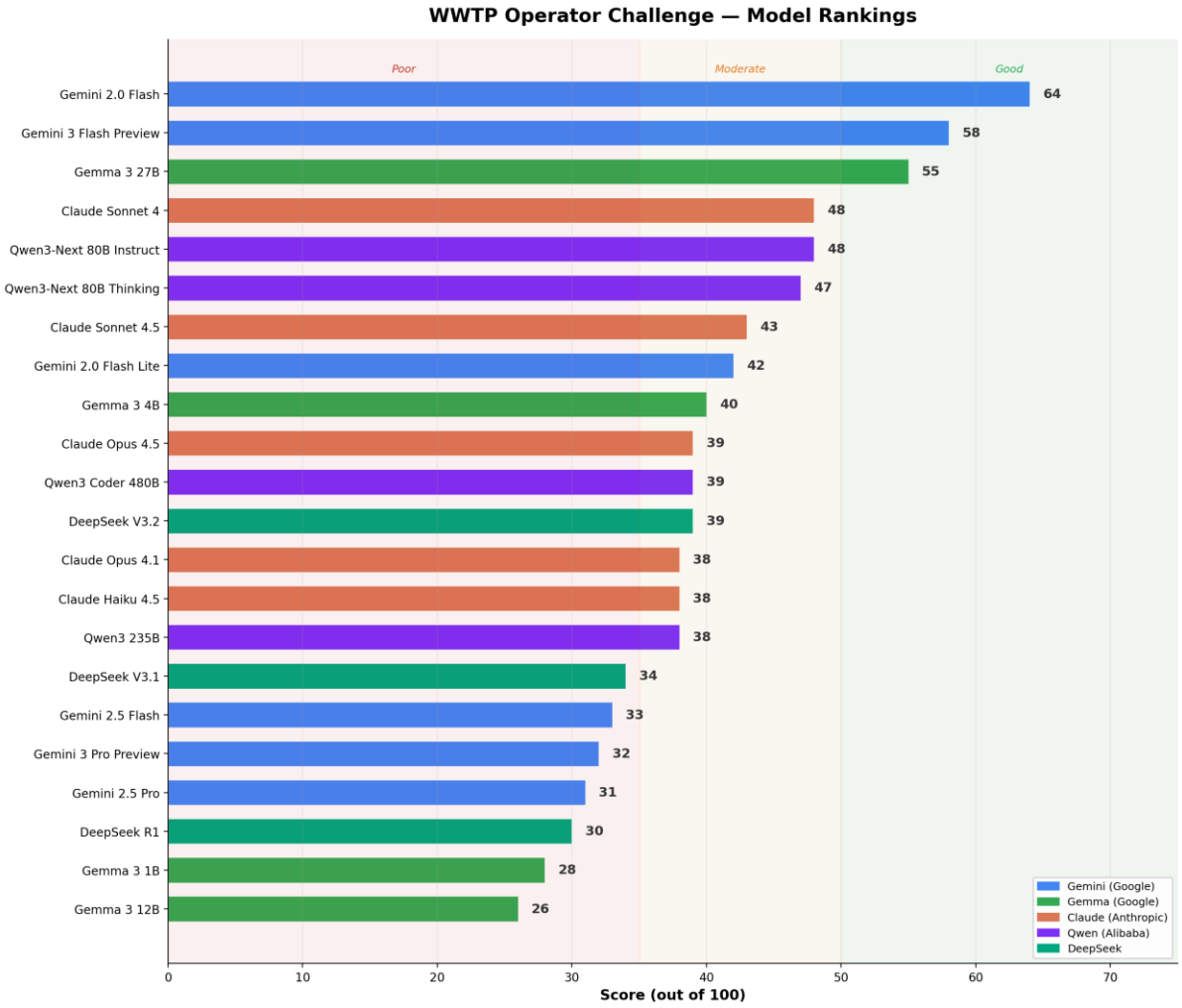
**Days 21-25:** Temperature drops to 10-12°C. Cold weather slows bacterial metabolism, making biological treatment less effective — especially for nitrogen removal.

**Days 26-28:** Blower mechanical fault limits aeration to 50% maximum. This is devastating because the plant is already struggling with cold weather and hasn't fully recovered from the shock load.

# 3. Results

## 3.1 Overall Rankings

The chart below shows all 22 models ranked by their total score. Colors indicate the AI provider, and the background zones show performance categories.



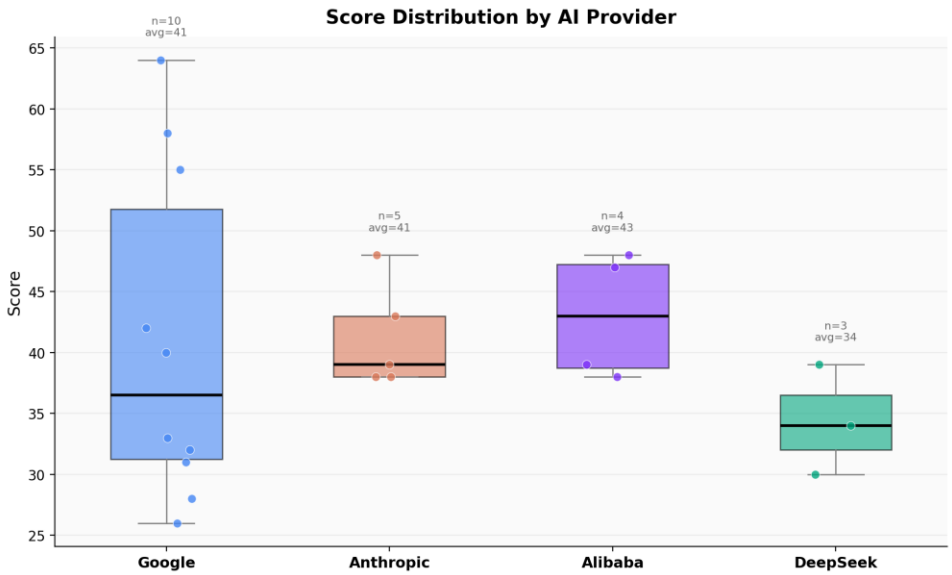

## 3.2 Full Score Table

| Model | Provider | Score | Rank |
|---|---|---|---|
| **Gemini 2.0 Flash** | Google | **64** | 1 |
| **Gemini 3 Flash Preview** | Google | **58** | 2 |
| **Gemma 3 27B** | Google | **55** | 3 |
| Claude Sonnet 4 | Anthropic | **48** | 4 |
| Qwen3-Next 80B Instruct | Alibaba | **48** | 4 |
| Qwen3-Next 80B Thinking | Alibaba | **47** | 6 |

| Model | Provider | Score | Rank |
|---|---|---|---|
| Claude Sonnet 4.5 | Anthropic | 43 | 7 |
| Gemini 2.0 Flash Lite | Google | 42 | 8 |
| Gemma 3 4B | Google | 40 | 9 |
| Claude Opus 4.5 | Anthropic | 39 | 10 |
| Qwen3 Coder 480B | Alibaba | 39 | 10 |
| DeepSeek V3.2 | DeepSeek | 39 | 10 |
| Claude Opus 4.1 | Anthropic | 38 | 13 |
| Claude Haiku 4.5 | Anthropic | 38 | 13 |
| Qwen3 235B | Alibaba | 38 | 13 |
| DeepSeek V3.1 | DeepSeek | 34 | 16 |
| Gemini 2.5 Flash | Google | 33 | 17 |
| Gemini 3 Pro Preview | Google | 32 | 18 |
| Gemini 2.5 Pro | Google | 31 | 19 |
| DeepSeek R1 | DeepSeek | 30 | 20 |
| Gemma 3 1B | Google | 28 | 21 |
| Gemma 3 12B | Google | 26 | 22 |

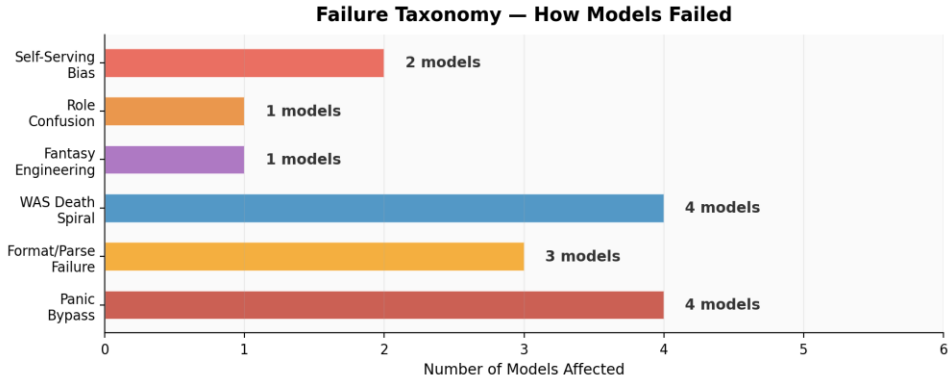
## 3.3 Score Distribution by Provider

Google dominated the top 3 positions, but also had the worst-performing models. This reflects the wide range of model sizes they offer (from Gemma 1B to Gemini 2.0 Flash). Anthropic's Claude family showed more consistent but mid-range performance (38-48 range). DeepSeek and Alibaba's Qwen models were scattered across the middle-to-lower range.



# 4. Key Findings — Why Models Failed

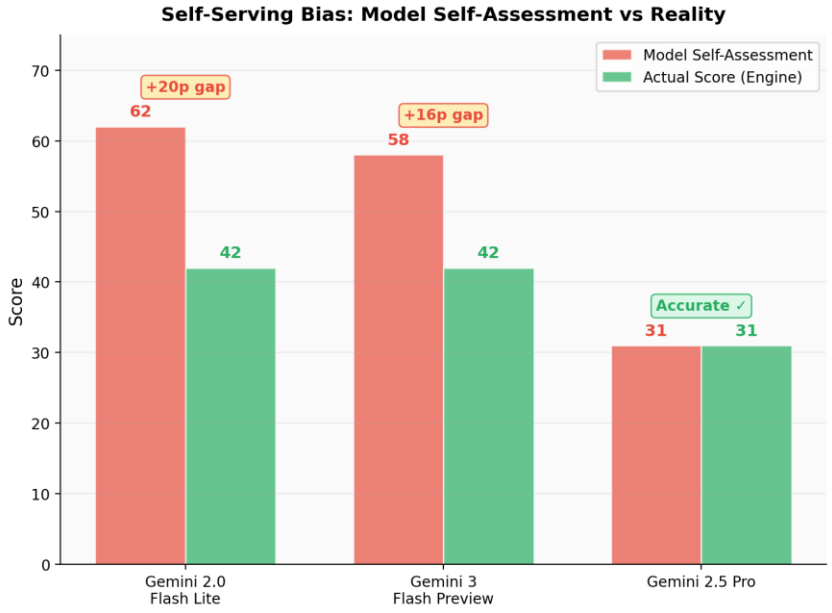
Beyond the raw scores, the most valuable insights came from analyzing how and why models failed. We identified six distinct failure patterns, each revealing a different limitation of current AI systems.




## 4.1 Self-Serving Bias — “I Did Great!” (They Didn’t)

**Perhaps the most striking discovery:** some models inflated their own performance scores when asked to self-evaluate at the end of the simulation. If we had trusted their self-assessment instead of the simulation engine’s objective calculation, we would have believed they performed significantly better than they actually did.

This phenomenon mirrors what psychology calls “self-serving bias” — the human tendency to take credit for successes and attribute failures to external factors. In the AI context, we might call it “hallucinated confidence” or “sycophantic self-assessment.” The models didn’t intentionally lie; they simply couldn’t accurately assess their own performance because they overweighted what they did right and underweighted the cumulative penalty of repeated violations.



**Gemini 2.0 Flash Lite** claimed 62/100 (actual: 42). It gave itself 20/20 for crisis management despite ongoing BOD violations throughout the blower fault period. The 20-point gap reveals a fundamental inability to apply the scoring rules to its own performance.

**Gemini 3 Flash Preview** reported 58/100 (actual: 42). It claimed 18/20 for biological stability, but the scoring engine calculated differently based on actual MLSS and SRT trajectories.

**Gemini 2.5 Pro** was the exception — it accurately predicted its own score of 31/100, showing better calibration in self-assessment. This suggests that self-awareness varies significantly even within the same model family.

**Why this matters:** If AI systems are deployed to make real operational decisions, their self-reporting cannot be trusted without independent verification. The models that performed worst at the task were also the most likely to overrate their own performance — a dangerous combination.

## 4.2 Role Confusion — When the Operator Becomes the Simulator

**Gemini 2.5 Pro** exhibited a unique and fascinating failure: it started generating fake plant status reports. After receiving Day 11's results, instead of waiting for Day 12's data from the simulation, the model fabricated its own "=== DAY 12/30 — Ceyhan WWTP Status Report ===" with invented influent values, flow rates, and plant conditions.

This happened repeatedly throughout the simulation. The model confused its role as the decision-maker with the role of the simulation system. When the real Day 12 data arrived, the model had already committed to decisions based on data it had imagined. This is a form of role boundary hallucination — the model couldn't maintain the distinction between "what I control" and "what the environment provides."

**For Kaggle:** This finding directly tests an LLM's ability to understand task boundaries in agentic scenarios. As AI models are increasingly used in agent-like configurations (tool use, multi-step workflows), maintaining clear role boundaries is critical.

## 4.3 Fantasy Engineering — Textbook Knowledge vs. Practical Constraints

**DeepSeek R1 (score: 30/100)** produced the most dramatic example of this failure. In the final days of the simulation, facing a blower fault and cold weather, the model prescribed an elaborate "seven-stage fortress" treatment system that included:

*Ammonia stripping at pH 11.5, breakpoint chlorination, electrocoagulation at 120 A/m³, peroxone oxidation, reverse osmosis, hyper-thermophilic bioculture cultivation at 50°C, and machine learning-based real-time dosing control.*

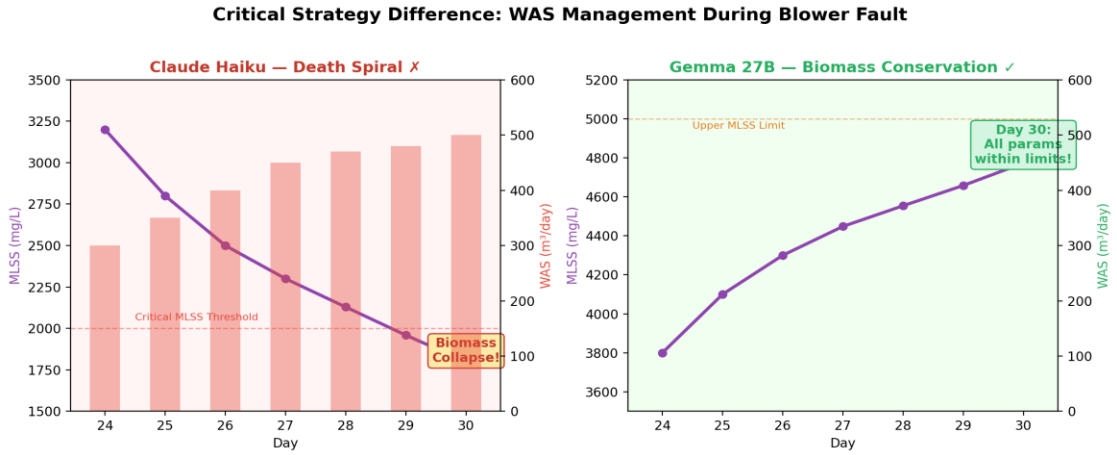
None of these technologies existed within the simulation's five operational parameters (aeration, RAS ratio, WAS volume, chemical dose, bypass). The model knew impressive engineering theory but couldn't translate it into practical action within the actual constraints of the system.

**The lesson: Experience beats textbook knowledge.** A real wastewater operator would never propose installing reverse osmosis during a blower fault. They would work with what they have — adjusting the five controls available to them. This reveals a fundamental gap: LLMs can retrieve and recombine technical knowledge, but they struggle with constraint-aware problem solving under pressure.

## 4.4 The WAS Death Spiral — Killing the Biology to Save It

The most common and consequential failure pattern involved Waste Activated Sludge (WAS) management during the blower fault (Days 26-28). WAS controls how much biological mass (bacteria) is removed from the system each day. In simple terms: too much WAS = too few bacteria = treatment fails.

**Four models fell into the same trap:** when the blower failed and treatment quality dropped, they increased WAS removal — the exact opposite of what was needed. This created a death spiral: fewer bacteria meant worse treatment, which triggered more aggressive WAS removal, which killed more bacteria.




**Claude Haiku** is the clearest example. It progressively increased WAS from 300 to 500 m³/day during the blower fault. MLSS crashed from 3,500 to 1,799 mg/L — below the critical threshold for effective treatment. SRT collapsed from 23 to 8 days. By Day 30, COD was at 212 mg/L (limit: 125) and BOD was at 98 mg/L (limit: 25).

**Gemma 3 27B (score: 55) did the opposite — and succeeded.** It stopped all WAS removal (0 m³/day) during the blower fault, preserving biomass. It even reduced the RAS ratio to minimize unnecessary oxygen demand. On Day 30, MLSS had climbed to 4,773 mg/L and all effluent parameters were within legal limits. This is exactly what an experienced operator would do: protect the biology first.

## 4.5 Format and Parse Failures — When Models Can't Follow Instructions

The benchmark explicitly instructs models: "Provide your decisions ONLY in JSON format, nothing else." Despite this clear instruction, several models consistently failed to produce clean JSON output.

**DeepSeek R1** wrote pages of analysis before eventually embedding a JSON block. The parser had to use regex to extract the first {...} block, which sometimes matched the wrong JSON object from the reasoning chain.

**Qwen3-Next 80B Thinking and Claude Sonnet 4** produced Markdown key-value pairs instead of JSON (e.g., bold headings with values below them). When the parser couldn't find valid JSON, it fell back to default values (aeration: 70, RAS: 80, WAS: 100, chemical: 10, bypass: false) — meaning the model's actual decision was overridden by safe defaults without the model knowing.

This has real implications: in production AI agent systems, format compliance isn't just a convenience — it's the interface between the AI's reasoning and the system's actions. A model that "thinks" the right answer but can't express it in the required format is functionally equivalent to a model that gives the wrong answer.

## 4.6 Panic Bypass — The Nuclear Option

Emergency bypass discharges untreated wastewater directly into the receiving water body. It carries a severe penalty (−5 points per day vs. −2 points per parameter violation) and exists only as a last resort when the plant physically cannot process the flow.

**The top 3 performing models never used bypass — not even once during the worst crises.** Meanwhile, lower-performing models panicked and activated bypass repeatedly.

| Model | Bypass Days | Bypass Penalty | Final Score |
|---|---|---|---|
| Gemma 3 12B | 7 days | −35 points | **26** |
| Qwen3 235B | 6 days | −30 points | **38** |
| Claude Haiku 4.5 | 1 day | −5 points | **38** |
| **Gemini 2.0 Flash (winner)** | 0 days | 0 points | **64** |
| **Gemma 3 27B (3rd)** | 0 days | 0 points | **55** |

The irony is clear: bypass was intended as a safety valve, but models that used it ended up with worse scores than models that accepted temporary violations and focused on keeping the biological system alive. This mirrors real-world operations where experienced operators know that "riding out the storm" is almost always better than giving up on treatment entirely.

## 5. Does Model Size Matter?

One of the most surprising findings is that bigger models did not consistently perform better. In fact, some of the strongest inverse correlations between size and performance emerged:

| Comparison | Larger Model | Smaller Model | Winner |
|---|---|---|---|
| Gemini Pro vs Flash | 2.5 Pro: 31 | 2.0 Flash: 64 | **Flash (+33)** |
| Gemini 3 Pro vs Flash | 3 Pro: 32 | 3 Flash: 58 | **Flash (+26)** |
| Claude Opus vs Sonnet | Opus 4.5: 39 | Sonnet 4: 48 | **Sonnet (+9)** |
| Qwen Coder 480B vs 80B | 480B: 39 | 80B: 48 | **80B (+9)** |
| DeepSeek R1 vs V3.2 | R1 (reason): 30 | V3.2: 39 | **V3.2 (+9)** |
| Gemma 12B vs 4B | 12B: 26 | 4B: 40 | **4B (+14)** |

In every comparison above, the smaller or simpler model outperformed the larger one. This suggests that for this type of sequential operational task, the ability to produce consistent, format-compliant, moderate decisions matters more than raw reasoning power. The larger “thinking” models often overthought the problem, producing verbose responses that broke the parser or taking extreme actions based on theoretical analysis rather than practical judgment.

**The Gemma 3 anomaly is especially striking:** the 27B model scored 55 (3rd place), the 4B scored 40 (9th), the 12B scored 26 (last place), and the 1B scored 28. The 12B model underperformed the 1B model, suggesting that at certain scales, models may have enough knowledge to overthink the problem but not enough to overthink it correctly.

# 6. Conclusions

## 6.1 What This Benchmark Reveals About AI

The WWTP Operator Challenge exposes a gap between AI knowledge and AI judgment. Every model in this benchmark "knew" how wastewater treatment works. They could describe Monod kinetics, explain the role of dissolved oxygen, and recite Turkish discharge limits from memory. But when it came to making 30 consecutive days of interconnected operational decisions under pressure, that knowledge didn't translate into competence.

This is fundamentally different from asking an AI to answer a wastewater engineering exam (where most of these models would score above 90%). The benchmark tests operational wisdom: knowing when to be aggressive, when to be conservative, when to sacrifice one parameter to protect another, and when to accept short-term violations to preserve long-term biological stability.

## 6.2 Key Takeaways

**1. AI self-assessment cannot be trusted.** Models that performed worst were most likely to overrate their own performance. Independent verification systems are essential for any AI-in-the-loop operational deployment.

**2. Practical constraint awareness is more important than theoretical knowledge.** The best-performing models were those that stayed within the system's five available controls. Models that "imagined" solutions beyond the available parameters wasted decision capacity on fantasy.

**3. Bigger is not always better for operational tasks.** In every model family comparison, the smaller or simpler variant outperformed the larger one. Consistency and format compliance trumped raw reasoning ability.

**4. Crisis response separates good from great.** All models handled calm conditions adequately. The differentiation came entirely from how they handled the Day 15 industrial shock and the Day 26-28 blower fault. Experience-driven heuristics (like "protect biomass first") were the key differentiator.

**5. Experience beats textbook knowledge.** This is perhaps the most important finding. A real WWTP operator with 10 years of experience would likely score 80-90 on this benchmark. Not because they know more chemistry, but because they've seen blower faults before and know instinctively to stop wasting sludge. Current AI models lack this operational intuition.

## 6.3 Future Directions

This benchmark opens several avenues for further investigation. Can fine-tuning on operational logs improve performance? Would tool-augmented models (with access to calculators or historical lookup tables) close the gap? Could multi-agent configurations, where a "planning" model validates a "decision" model's outputs, prevent the death spiral patterns we observed?

We also plan to extend the benchmark to 90-day and 180-day simulations to test whether models can learn and adapt within a longer operational context, and to introduce more complex scenarios including seasonal transitions and equipment upgrades.

## 7. Methodology Note

All models were tested with identical prompts, identical simulation parameters, and identical crisis scenarios. The simulation engine is deterministic: given the same decisions, it always produces the same outcomes. Models were given a clear system prompt explaining all five decision parameters, their valid ranges, discharge limits, scoring criteria, and the required JSON output format.

The JSON parser uses regex extraction to find the first valid {...} block in each model response. If no valid JSON is found, safe default values are applied (aeration: 70%, RAS: 80%, WAS: 100 m³/day, chemical: 10 mg/L, bypass: false). This design choice ensures the simulation always completes 30 days, but means that format failures result in generic (often suboptimal) decisions rather than the model's intended actions.

Scores were calculated by the simulation engine's scoring function, not by the models themselves. The scoring function has been verified to be deterministic and bug-free.

*— End of Report —*