1.2 Cost-Performance Analysis

The document reveals a clear correlation between performance and cost, especially for reasoning-intensive tasks:

Model	ARC-AGI-1 Score	Avg Tokens	Approx. Cost per Query
GPT-4o	5%	1,000	$0.03
DeepSeek-R1	15.8%	6,000	$0.06
o1-mini	15.6%	7,000	$0.11
o1 (low)	20.5%	7,000	$0.43
o1 (med)	31%	13,000	$0.79
o1 (high)	35%	22,000	$1.31
o3 (low)	75.7%	335,000	$20.00
o3 (high)	87.5%	57M	$3,400.00

1.3 Current Pricing Analysis

Looking at the OpenAI pricing table, we observe:

Model	Input (per 1M tokens)	Output (per 1M tokens)
gpt-4o	$2.50	$10.00
gpt-4o-mini	$0.15	$0.60
o1	$15.00	$60.00
o1-mini	$1.10	$4.40
o3-mini	$1.10	$4.40

3. Strategic Recommendations for Turing Labs

3.1 Tiered Model Selection Strategy

Based on our analysis, we recommend implementing a tiered model selection approach:

1. **Base tier (80% of queries)**: Use gpt-4o-mini for simple, well-defined tasks like information extraction, summarization, and basic content generation. Cost: ~$0.15-0.60/1M tokens.
2. **Mid tier (15% of queries)**: Deploy o1-mini or o3-mini for tasks requiring moderate reasoning such as complex data analysis, code generation, and problem-solving in familiar domains. Cost: ~$1.10-4.40/1M tokens.
3. **Premium tier (5% of queries)**: Reserve o1 for mission-critical tasks requiring high reliability such as financial analysis, complex business logic implementation, and novel problem-solving. Cost: ~$15.00-60.00/1M tokens.

3.2 Dynamic Model Routing System

We recommend implementing an intelligent router that:

1. Analyzes incoming queries to predict reasoning complexity
2. Routes queries to the appropriate model tier based on:
    - Task complexity
    - Required reliability
    - Customer SLA requirements
    - Cost sensitivity

This would maximize cost-efficiency while maintaining performance standards.

3.3 Strategic Considerations for DeepSeek-R1 Integration

The open-source nature of DeepSeek-R1 presents an opportunity for Turing Labs to:

1. Self-host the model for certain workloads to potentially reduce costs
2. Fine-tune the model on company-specific domains to enhance performance
3. Create specialized versions for specific customer industries

However, this requires weighing infrastructure and operational costs against API costs.

3.4 Performance Monitoring and Optimization

We recommend implementing:

1. A continuous evaluation framework to track model performance across different task types
2. A feedback loop to identify when tasks are unnecessarily routed to higher-tier models
3. Cost optimization algorithms that balance performance and expense

3.5 Pricing Strategy for Turing Labs' Product

Based on the reliability-cost relationship, we recommend:

1. **Tiered pricing plans**:
    - Standard: Primarily using base tier models
    - Professional: Access to mid-tier models for complex tasks
    - Enterprise: Full access to premium models for mission-critical applications
2. **Usage-based surcharges** for premium model access beyond plan allocations
3. **Performance guarantees** tied to pricing tier (higher tiers receive stronger reliability commitments)