

Model selection begins with the workload

Choose a model by evaluating the application's operating requirements.

Evaluate

- Required input and output modalities
- Task complexity
- Expected response quality
- Context-window requirements
- Latency and throughput targets
- Cost per request
- Safety and deployment requirements

Key message

A benchmark leader may perform poorly against an application's operating constraints.

Deployment makes a model available to applications

A deployment provides application access to a selected model.

Flow · “Application request → Model endpoint → Inference → Response”

The request contains

- Authentication
- A deployment or model identifier
- Input content
- Generation settings

The endpoint returns

- Response: The model’s generated output from inference

Remember

Deployment prepares a model for inference. Model choice and deployment choice are separate decisions.

Availability depends on model, region, and deployment type

Before selecting a model, confirm current availability—not just the catalog listing.

Confirm

- The model is available in the required Azure location.
- The region supports the required deployment type.
- Sufficient quota or capacity is available.
- Data processing locations meet organizational requirements.
- The required model version has an acceptable lifecycle status.

Key message

Availability can differ by model, model version, region, and subscription.

Shared and dedicated capacity

Capacity approach	Conceptual behavior	Best fit
Shared capacity	Customers use a common pool of model-processing capacity	Variable traffic, evaluation, and general workloads
Dedicated capacity	The deployment reserves processing capacity for one customer	Sustained demand and predictable performance

Typical tradeoff

Shared
Often uses consumption-based pricing.

Dedicated
Improves throughput predictability.

Reservation
May incur cost even during low usage.

Throughput, latency, cost, and quality

These four measures compete. Optimize for the workload, not a single score.

Measure

Measure	Question
Throughput	How much work can the deployment process over time?
Latency	How long does one request take to return a response?
Cost	What does inference capacity or consumption cost?
Quality	How well does the model perform the required task?

Examples

- A live voice assistant needs low latency.
- Batch document processing prioritizes throughput and cost.
- A complex legal analysis may justify a slower, stronger model.
- High-volume classification may favor a smaller model.

Remember

A deployment can have strong throughput while an individual request still takes several seconds.

Model size introduces tradeoffs

Parameter count alone does not determine application value.

Larger models may provide

- Better performance on difficult or unfamiliar tasks
- Broader knowledge and instruction-following ability
- Stronger handling of nuanced inputs

Smaller models may provide

- Lower latency
- Lower cost
- Higher throughput
- Sufficient quality for focused tasks

Key idea

The smallest model that meets the quality requirement often provides the best operational fit.

Deployment decision scenario

“A support assistant receives thousands of routine questions each day. Most answers come from company documentation. Customers expect quick responses, and the company must control cost.”

Requirement	Design response
Current company information	Ground responses with retrieved documentation
Fast routine answers	Use a lower-latency model
Complex exceptions	Route selectively to a stronger model
Variable early traffic	Use shared consumption capacity
Stable high traffic	Evaluate dedicated capacity

Recommended approach

- 1. Use an embedding model to retrieve relevant documentation.
- 2. Use a capable smaller language model for routine answers.
- 3. Route difficult cases to a stronger reasoning model.
- 4. Begin with shared capacity while traffic remains variable.
- 5. Consider dedicated capacity when demand becomes sustained and predictable.

Selection logic

Routing reserves expensive capabilities for the requests that need them.