Adaptive T&E via Bayesian Decision Theory
DATAWorks 2024

Jim Ferry, Nate Crookston, and Adam Ahmed

April 18, 2024

Sponsors:
Dr. Sandra Hobson, DOT&E, Deputy Director, SIPET
Dr. Jeremy Werner, DOT&E, SIPET, Chief Scientist

COR:
Mr. Chris Dodson, DOT&E JT&E

How much is conducting a single trial of a system worth?

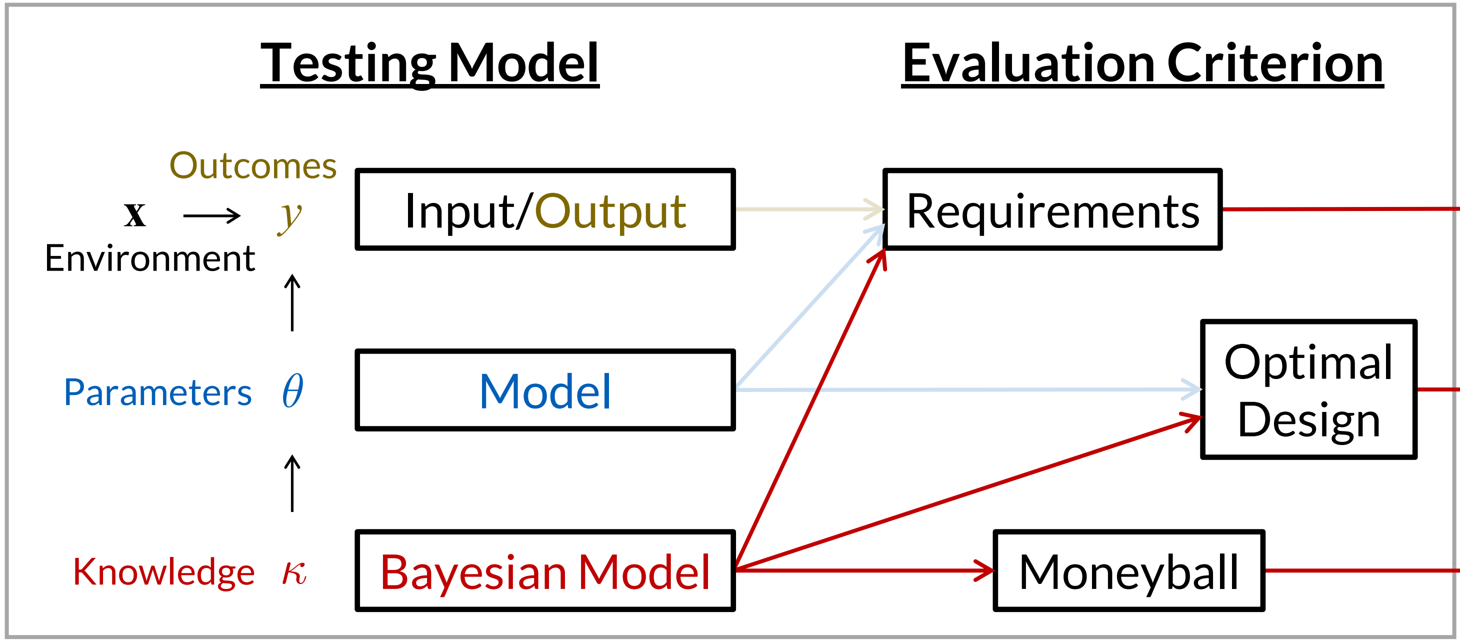
Executive Summary

- Bayesian Sequential Testing
  - Bayesian model maintains *knowledge* about system under test
    - Enables knowledge to be ported between test events
    - Predicts impact of even a single trial on knowledge about system
  - Leverages multiple evaluation criteria
    - Requirements: visualize progress toward meeting requirements
    - Optimal design: generate test design candidates
    - Moneyball: novel criterion for Bayesian models
- Moneyball evaluation criterion
  - Based on operational utility of system given current knowledge
    - Captures stakeholder priorities
    - Formulated in same units as testing cost: *enables principled cost/benefit analysis*
  - Recommends which trials are best, or whether it's time to stop testing

Dynamo T&E

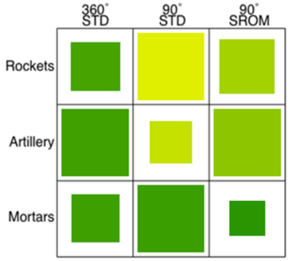
- Dynamic Knowledge via Bayesian model of system
- Moneyball and other evaluation criteria

Dynamo T&E: *Dyna*mic Knowledge + *Mo*neyball



Application in Dynamo T&E

- Visualize state of test



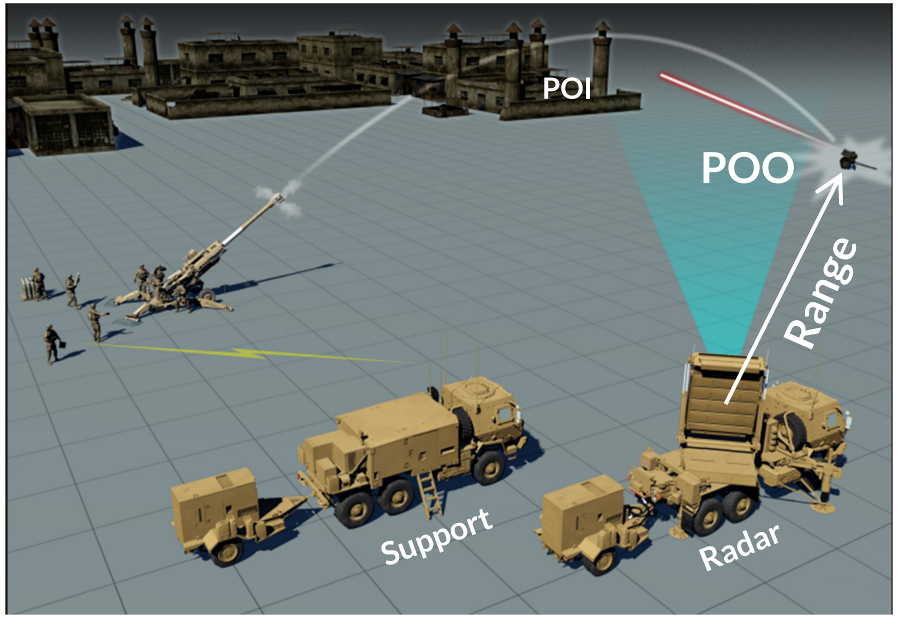
- Generate candidate test designs
- Real-time T&E decision aid
- Recommends
  - Which environments $\mathbf{x}$ to test
  - When testing is no longer worthwhile

- Dynamic Knowledge
  - Bayesian model
  - Ports knowledge between test events
  - Real-time decisions
- Moneyball evaluation
  - All stakeholders' priorities put into common currency of operational utility
  - Subsumes Requirements and Optimal Design criteria
  - Includes cost of testing

AN/TPQ-53 : Exemplar for Tabletop Demo

- Demo based on data from IOT&E 2 test event
  - Held at Yuma Proving Grounds, summer 2015
- Demo provides example of a decision-support tool for a dynamic test event
  - In contrast to a static test design, which cannot incorporate the results of test
- Exemplar system: AN/TPQ-53
  - Estimates **Point Of Origin** (**POO**) and Point Of Impact
  - POO provided to counterfire shooters
  - Detects projectiles in flight while scanning 90˚ or 360˚ search area
    - Can detect projectiles of varying aspect angles: incoming, crossing, etc.
  - Characterizes in-flight projectiles as Mortar, Artillery, or Rocket

Lockheed Martin AN/TPQ-53



Environment

- Range
- Op Mode
- Aspect Angle
- Munition

First Step: Define Inputs and Outputs

Testing Model

Outcomes

$\mathbf{x} \longrightarrow y$ Input/Output

Environment

- Can collect input/output pairs $(\mathbf{x},y)$ for data analysis
- Test design?
  - Which environments $\mathbf{x}$ to test?
- Test evaluation?
  - Which outcomes y indicate system is good or bad in environment $\mathbf{x}$?

- **Define inputs and outputs**
  - Input: environment $\mathbf{x}$
    - Conditions under which a trial is made
    - E.g., range to target, depth, system configuration
  - Output: outcomes y
    - Results of a single trial
    - E.g., hit/miss, miss distance, time to failure

TPQ-53 case

- Environment $\mathbf{x}$ = munitions type, operating mode (90° vs 360°), radar-to-battery range, etc.
- Outcome y = Point Of Origin error between actual and estimated location of battery

Evaluation Criterion: Meeting Requirements

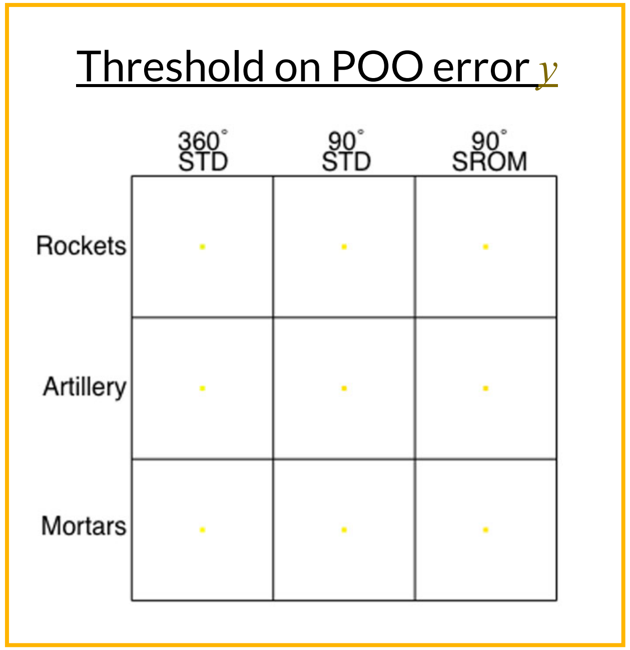
Testing Model — **Evaluation Criterion**

Outcomes
$\mathbf{x} \longrightarrow y$
Environment

Input/Output → Requirements

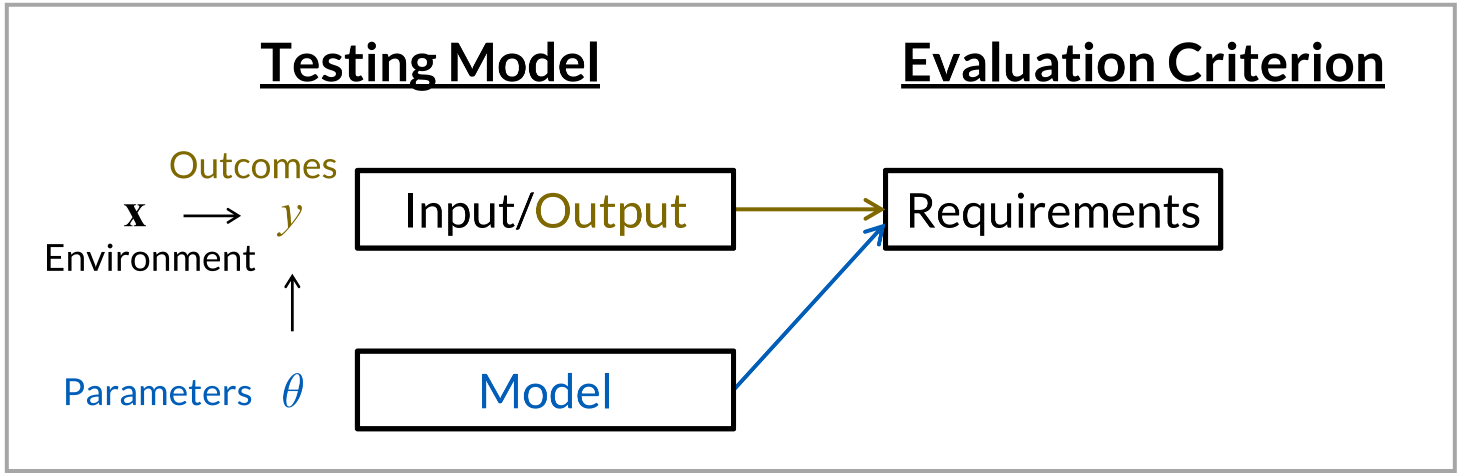
- (U) Define which outcomes are considered good

- Typical structure for specifying requirements
  - Group environments $\mathbf{x}$ into sectors
  - Set thresholds on y for each sector
  - Specify what fraction of outcomes must meet each threshold
  - TPQ-53: 9 sectors with thresholds for each →



- (U) Test design?
- (U) Prediction?
  - (U) What are outcomes for an environment that wasn't tested?

System Model Predicts System Performance



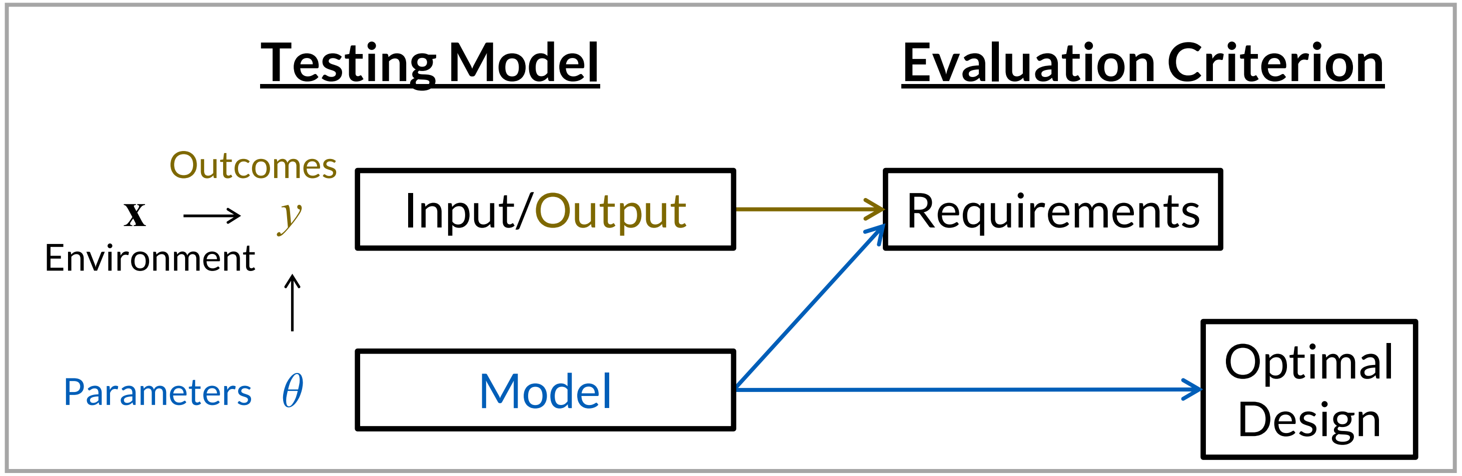
- Model: predicts outcomes y in any environment $\mathbf{x}$

- Model specifies how likely outcomes y are for any $\mathbf{x}$
  - Design system model with parameters θ: "tunable knobs"
  - Estimate θ from test data
- Predicts outcomes y in untested environments $\mathbf{x}$
- *Compliance fraction* in each sector (for a given θ)
  - Fraction of outcomes y within threshold over all $\mathbf{x}$ in sector

Compliance Fraction

	360° STD	90° STD	90° SROM
Rockets	0.79	0.54	0.61
Artillery	0.80	0.56	0.65
Mortars	0.81	0.81	0.86

Evaluation Criterion: Design of Experiments Metrics

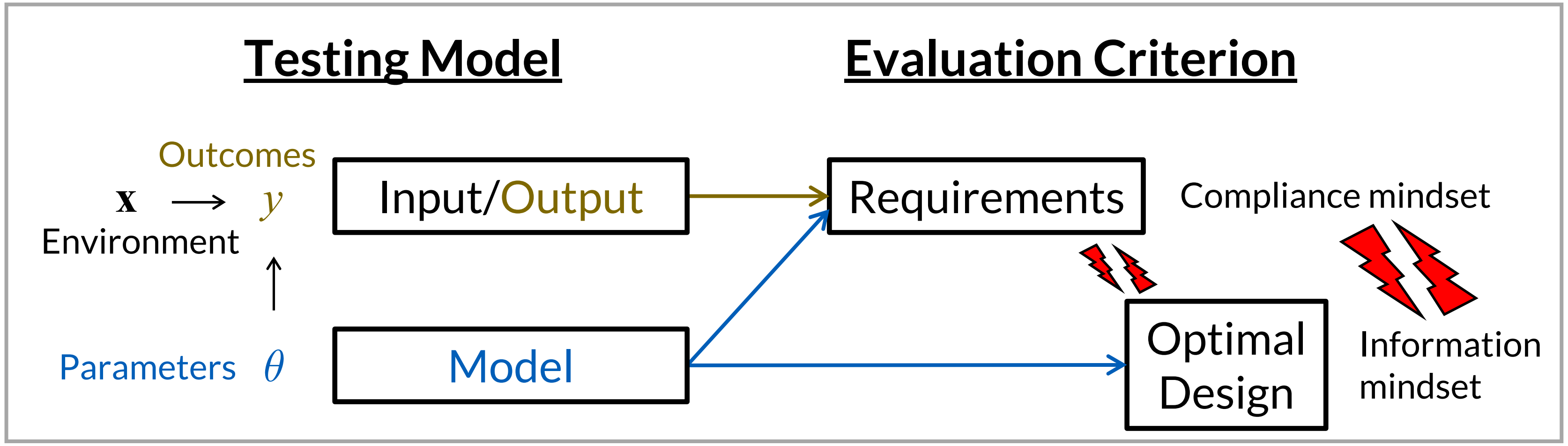


- **Optimal Design: select test environments $\mathbf{x}$ to optimize information about parameters θ**

- Theory of Optimal Design
  - Specifies which environments $\mathbf{x}$ to test
  - Goal: optimize some DoE metric for estimating parameters θ ⟶
- Benefit: optimal test design independent of outcomes y
  - Provides good estimate of θ regardless of system being good or bad
- Drawback: ignores requirements

- "Alphabet soup" of Design of Experiments (DoE) metrics:
  - A-optimality
  - C-optimality
  - D-optimality
  - E-optimality
  - S-optimality
  - T-optimality
  - G-optimality
  - I-optimality
  - V-optimality

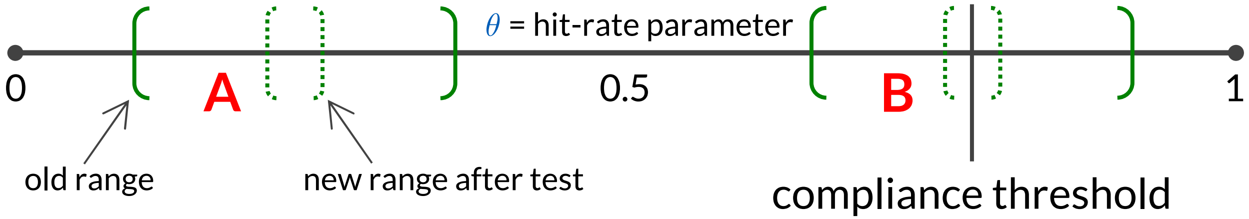
Interlude: What is the Goal of T&E?



- Is the goal of T&E:
  - To assess compliance?
  - Or to gain information?

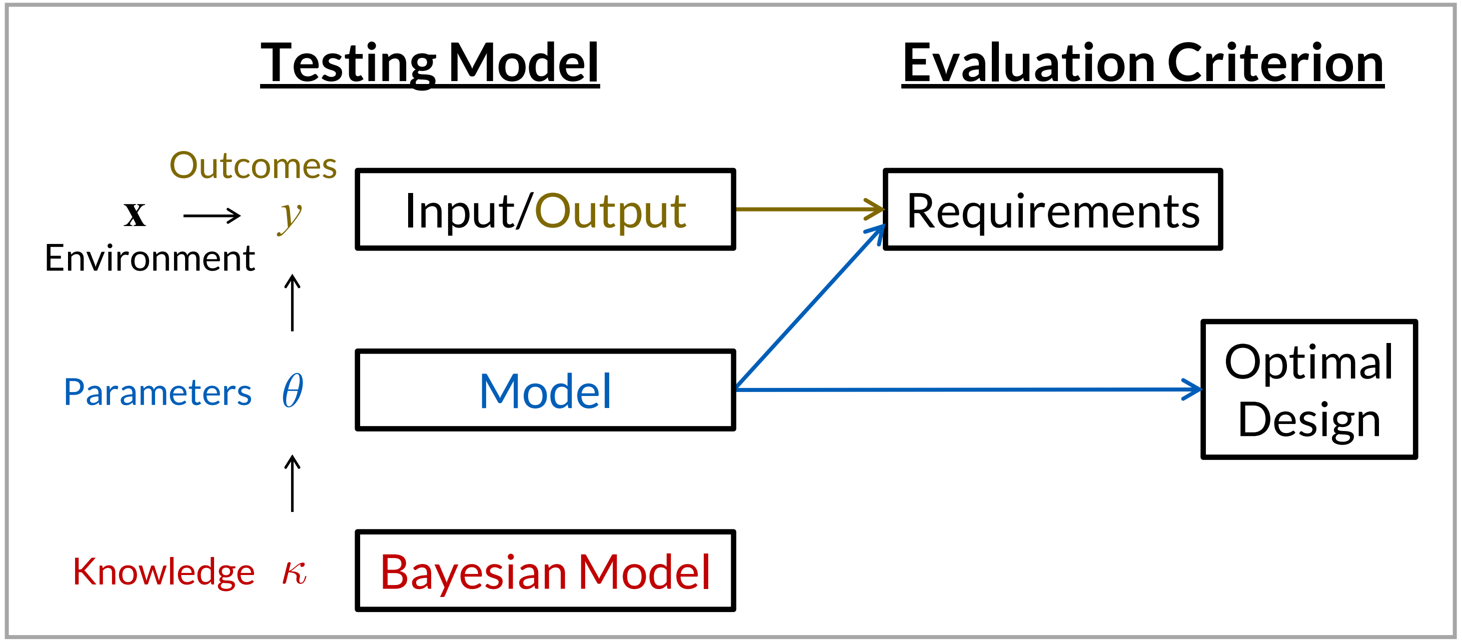
- Information mindset: to gain information about system most efficiently
  - Optimizes test design for precision regardless of test outcomes
  - Sees value in tightening estimate in case **A**: information is gained

- Compliance mindset: to determine whether system meets requirements
  - Only concern is confidence about system meeting requirements
  - Sees no value in tightening estimate in case **B**: still 50% chance of compliance



J. Ferry *et al.*, "Use of Bayesian Methods to Optimize Decisions," *Naval Engineers Journal* **136**(1), 2024 (in press)

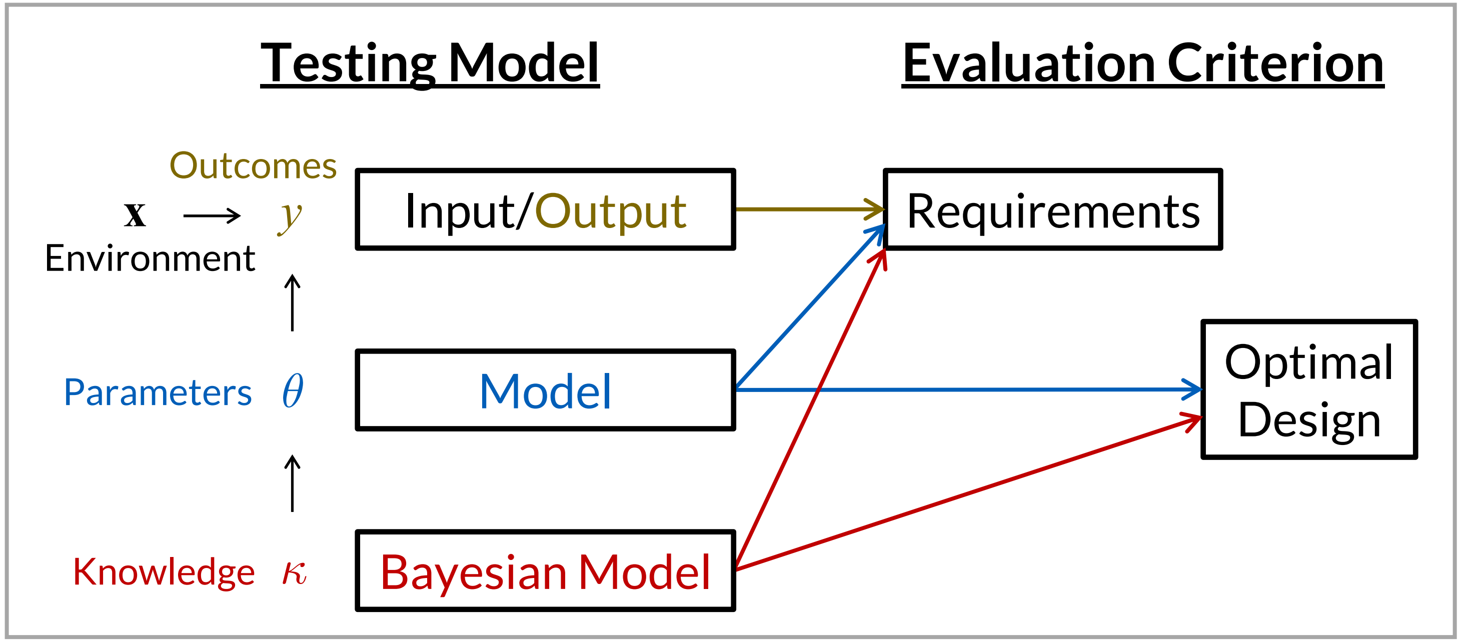
Bayesian Model



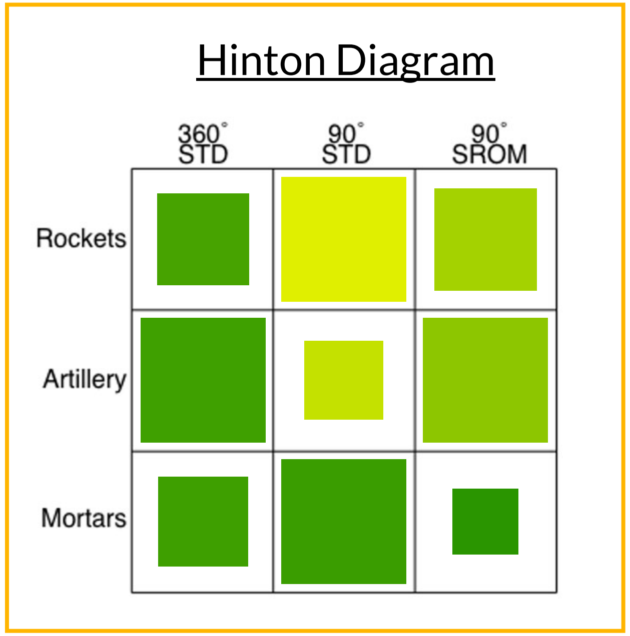
- Classical approach: estimate parameters θ
  - Done in batch after test event complete
- Bayesian approach: maintain knowledge κ about θ
  - Initialize κ with expert input and prior test event results
  - Update κ with each trial: outcomes y for environment $\mathbf{x}$

- **Bayesian Model: Maintains knowledge κ about parameters θ from experts and test data**
- Benefits of Bayesian approach
  - Can port knowledge between test events
  - Real-time decisions during test
  - Improved evaluation

Improved Evaluation with Bayesian Model

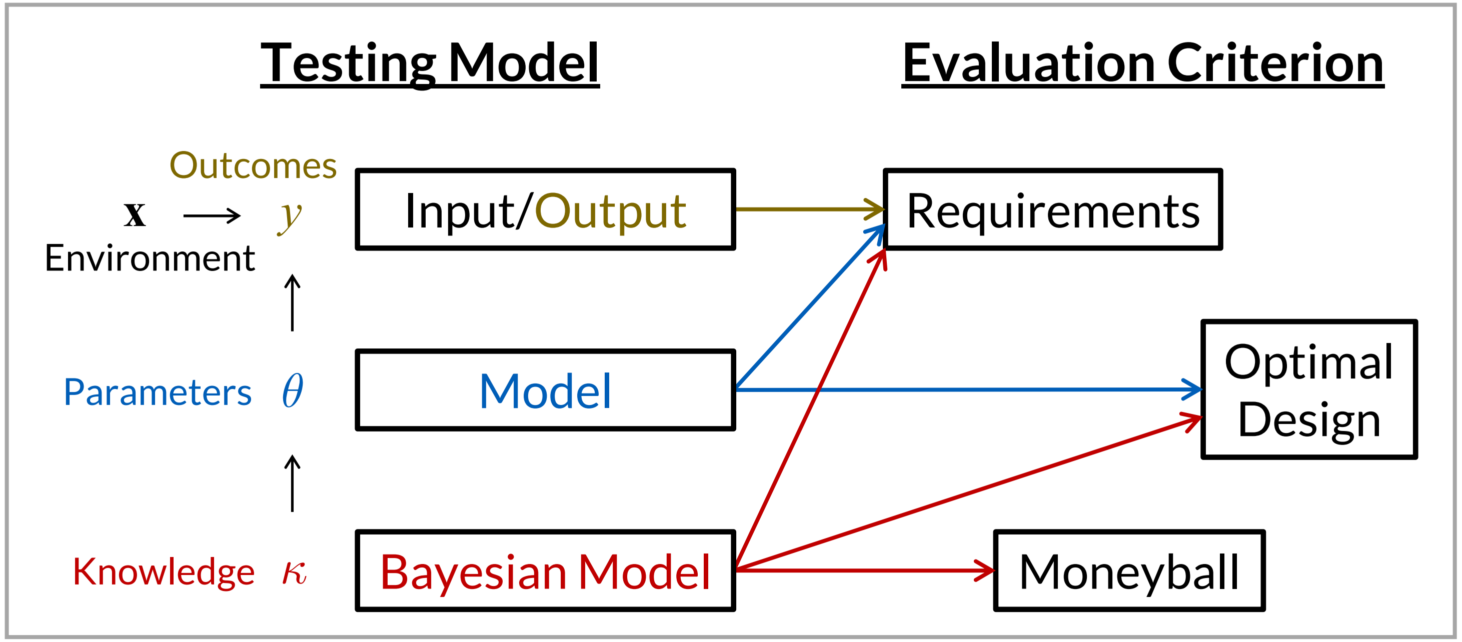


- Gives better assessment of
  - Whether requirements met
  - Optimal test design



- Bayesian assessment of whether requirements met
  - *Compliance fraction* (per sector) defined for any parameters θ
    - Precise knowledge κ about θ depicted as large box
    - Imprecise knowledge κ about θ depicted as small box
- Bayesian Experimental Design: leverage expert knowledge

Evaluation Criterion: Moneyball

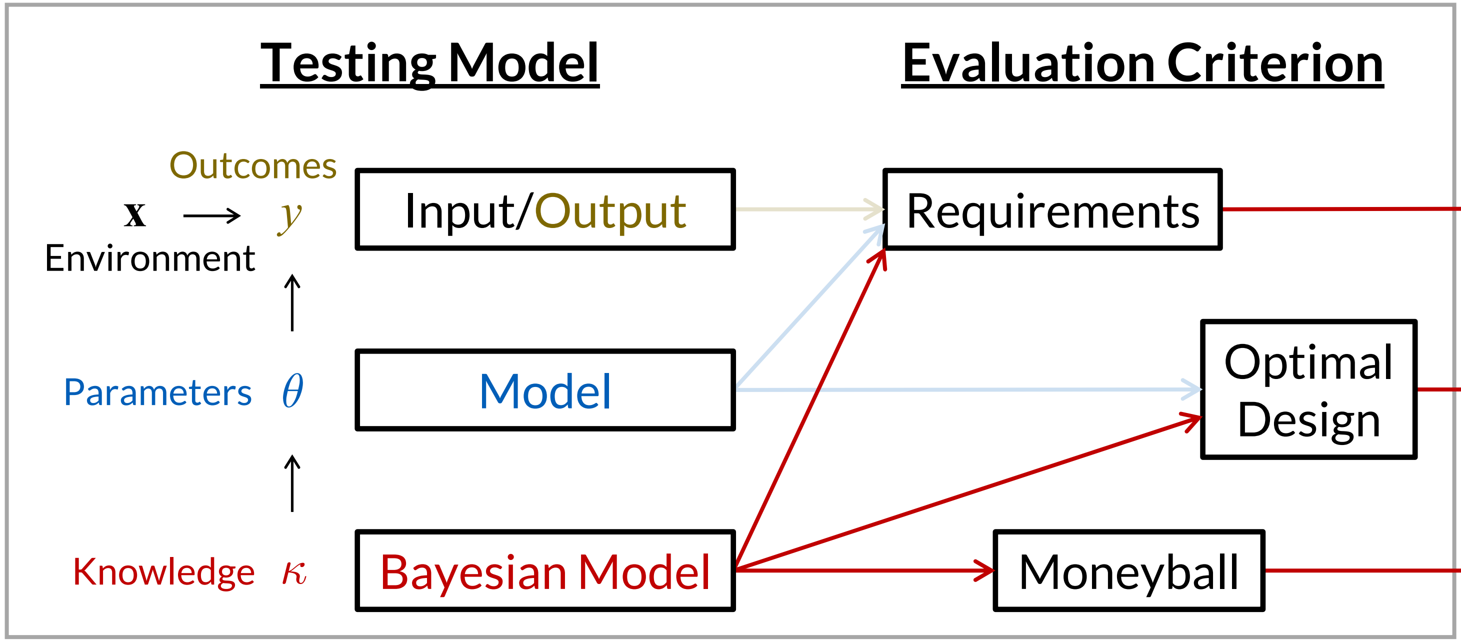


- Moneyball evaluation
  - Direct assessment of κ
  - All stakeholders' priorities put into common currency
  - Subsumes Requirements and Optimal Design criteria
  - Includes cost of testing

- Moneyball: a new evaluation criterion for Bayesian models
  - Define the operational *utility* of a system when knowledge about it is κ
  - Utility can be based on requirements, but include softer thresholds
  - Utility can represent the value of information by modeling its impact on operational decisions
- Testing decisions: weigh benefit of knowledge gain vs. cost of test

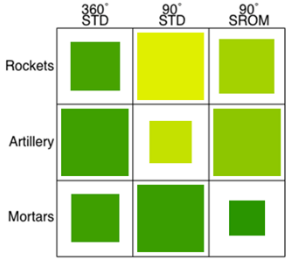
M. Lewis, *Moneyball: The Art of Winning an Unfair Game*, W. W. Norton and Company, 2003

Dynamo T&E



Application in Dynamo T&E

- Visualize state of test



- Generate candidate test designs
- Real-time T&E decision aid
- Recommends
  - Which environments $\mathbf{x}$ to test
  - When testing is no longer worthwhile

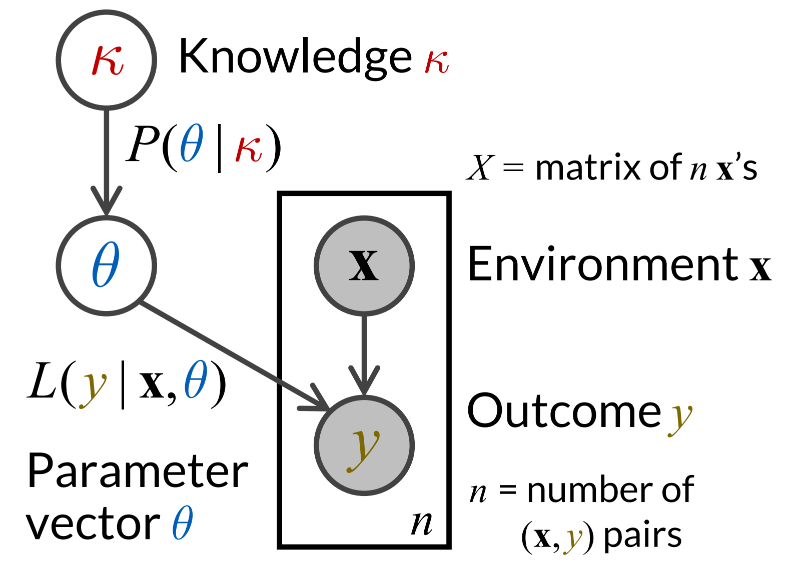
- Dynamo combines
  - A Bayesian model of knowledge that updates in real time
  - A Moneyball utility function that assesses decisions in terms of operational impact
- But how does it actually work?

Mathematical Structure

- Properties of Knowledge
- Governing equations for Utility
- Intrinsic utility: cares only about correct terminal decision
  - Impetus to test propagates out from terminal decision boundaries

Knowledge

- What does knowledge of a system mean?
  - There's knowledge in the TEMP, IDSK, test data, SMEs, etc.
  - Dynamo requires three properties of knowledge κ
- Property #1: κ provides distribution on y for any $\mathbf{x}$
  - E.g.: sample θ given κ then sample y given θ and $\mathbf{x}$
  - Or: use explicit formula $P(y \mid \mathbf{x}, \kappa) = \int L(y \mid \mathbf{x}, \theta) P(\theta \mid \kappa) d\theta$
- Property #2: can update κ to κ^+ as data $(\mathbf{x}, y)$ arrive
  - E.g.: represent κ as an ensemble of θ's and update using MCMC (Markov Chain Monte Carlo)
  - Or: represent κ as a hyperparameter in a conjugate prior family and update it directly
- Together, #1 and #2 provide $P(\kappa^+ \mid \kappa, X)$ for any matrix X of n environments to test



The knowledge κ characterizes how the system behaves in any environment $\mathbf{x}$

Utility

- Property #3: a *utility function* $u_d(\kappa)$ over κ is defined for all *terminal decisions* $d \in D$
  - Terminal decisions $d \in D$: d = Reject system, d = Accept system, d = Improve system, etc.
  - Utility $u_d(\kappa)$: expected benefit of terminal decision d when knowledge is κ
    - E.g., if d = Accept into Full-Rate Production, $u_d(\kappa)$ = value to military minus costs (production, etc.)
- Elicitation challenging because method makes all assumptions explicit
- Governing equations define a Sequential Bayesian Decision Theory problem

$$u(\kappa) \doteq \max\left(u_D(\kappa), u_C(\kappa)\right)$$ Utility of κ with options to stop or continue testing

$$u_D(\kappa) \doteq \max_{d \in D} u_d(\kappa)$$ Terminal utility of κ: no further testing allowed (Property #3)

$$u_C(\kappa) \doteq \max_{X \in C} u_X(\kappa)$$ Utility of best choice X for environments to test next

$$u_X(\kappa) \doteq \mathbb{E}_{\kappa^+|\kappa,X}\left[u(\kappa^+)\right] - c_X$$ Expected utility of testing environments X (Properties #1 and #2)

(c_X: testing cost for X)

Intrinsic Utility and Martingales

- Requirements, stakeholder preferences, etc. encoded in utility functions $u_d(\kappa)$
  - Important to understand the structure of $u_d(\kappa)$
- Every $u_d(\kappa)$ determines a $u_d(\theta)$ as a special case
  - I.e., when the knowledge κ = "precise value of θ known"
- Every $u_d(\theta)$ defines a certain type of $u_d(\kappa)$: an *intrinsic utility* $u_d^I(\kappa) \doteq \mathbb{E}_{\theta|\kappa}\left[u_d(\theta)\right]$
- Every $u_d(\kappa)$ can be decomposed into $u_d(\kappa) = u_d^I(\kappa) - c_d(\kappa)$
  - $u_d^I(\kappa)$ is about making the best terminal decision, on average
  - The *cost of imprecision* $c_d(\kappa)$ is the penalty for imprecise knowledge
- Intrinsic utilities form *martingales*: $\mathbb{E}_{\kappa^+|\kappa,X}\left[u_d^I(\kappa^+)\right] = u_d^I(\kappa)$
  - For any given $d \in D$ the utility $u_d^I(\kappa)$ is the same, on average, as its future value… so why test?
  - Because $\mathbb{E}_{\kappa^+|\kappa,X}\left[u_D^I(\kappa^+)\right] \geq u_D^I(\kappa)$ (with $u_D^I(\kappa) \doteq \max_{d\in D} u_d^I(\kappa)$)

<u>Alethophobia</u>: when $c_d(\kappa) < 0$ a perverse "fear of truth" can arise, where a tester would rather not know test results, even if they were free!

What does this say about a case with only one terminal decision?

Marginal Utility of the Option to Test

- For intrinsic utility, impetus to test generated at terminal decision boundaries (i.e., which $d \in D$ yields largest $u_d(\kappa)$) →
- To see this, re-write equations using

$$v(\kappa) \doteq u(\kappa) - u_D(\kappa) = \max\left(0, v_C(\kappa)\right)$$

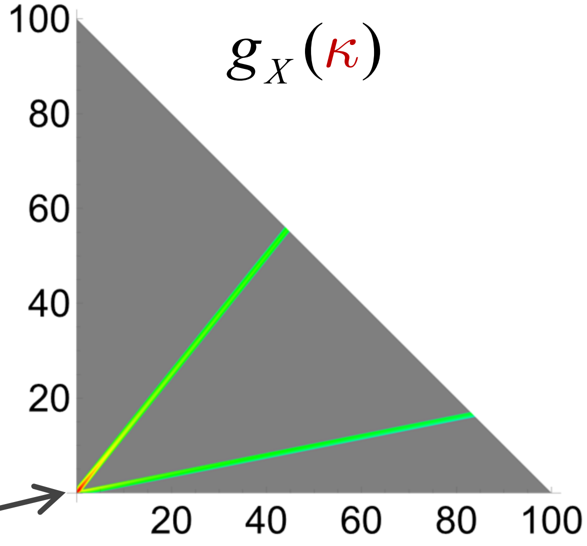
- New governing equations:

$$v_C(\kappa) \doteq \max_{X \in C} v_X(\kappa)$$

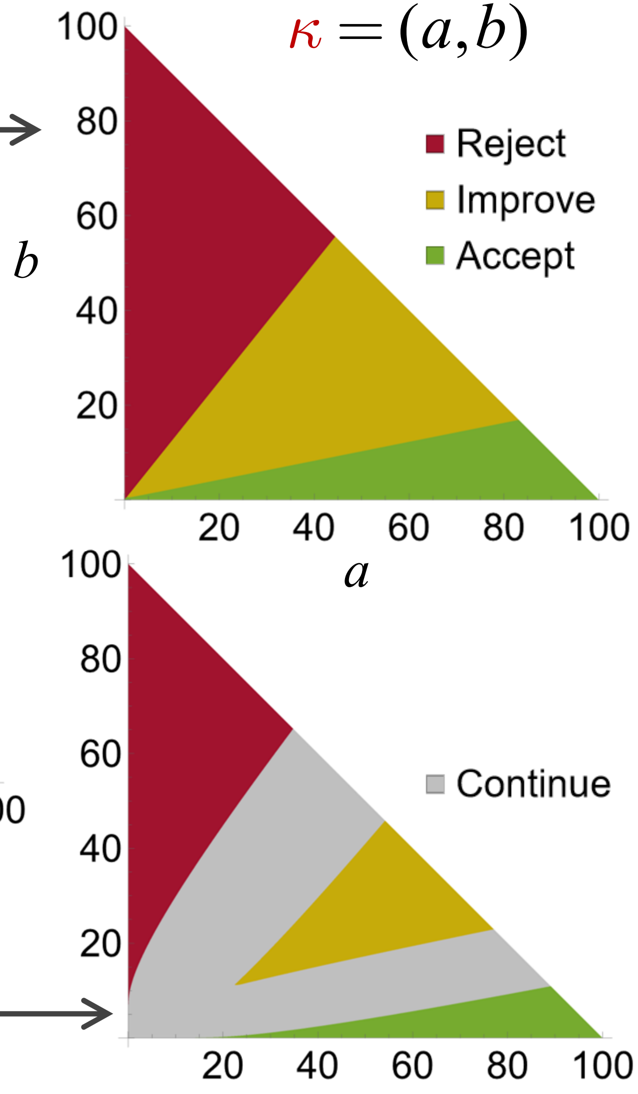
$$v_X(\kappa) \doteq \mathbb{E}_{\kappa^+|\kappa,X}\left[v(\kappa^+)\right] + g_X(\kappa) - c_X$$

- Source term at decision boundaries:

$$g_X(\kappa) \doteq \mathbb{E}_{\kappa^+|\kappa,X}\left[u_D(\kappa^+)\right] - u_D(\kappa)$$



- Impetus to test propagates out from decision boundaries →



Example: Simple Hit/Miss System

- Beta–Bernoulli model for dynamic knowledge
- Utility model with three terminal decisions
- Source of marginal utility at terminal decision boundaries
  - Determines Continue Testing regions

Bayesian Model for Hit/Miss System

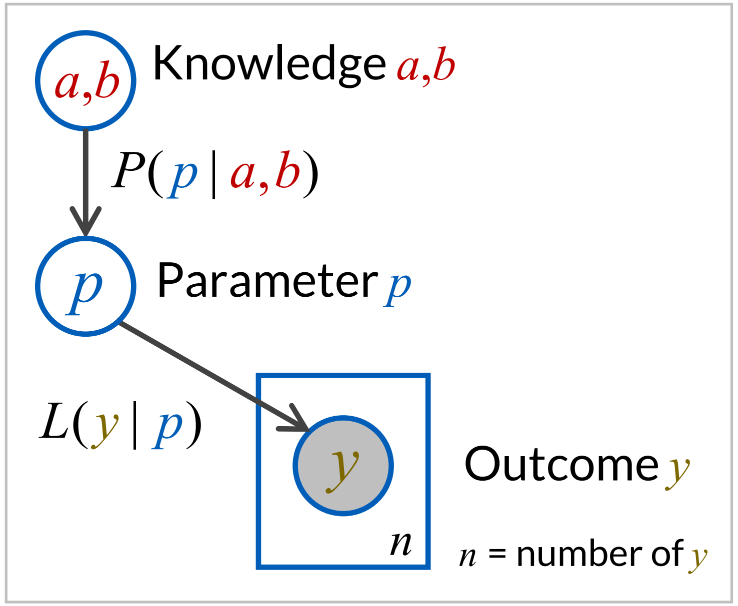
- Specialize equations to simple hit/miss case
  - No environment $\mathbf{x}$
  - Outcome y = 1 (hit) or 0 (miss)
  - Parameter vector $\theta = p$ (hit probability)
  - Knowledge $\kappa = (a,b)$

Beta distribution on p

$$P(p \mid a,b) = \frac{p^{a-1}(1-p)^{b-1}}{\mathrm{B}(a,b)}$$

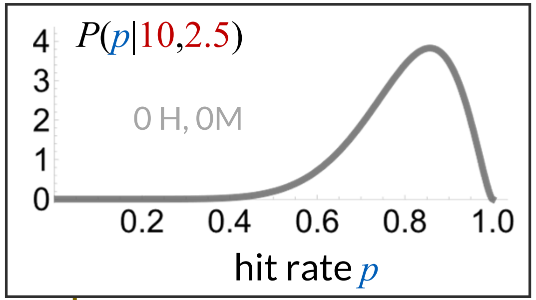
Bernoulli distribution on y

$$L(1 \mid p) = p$$
$$L(0 \mid p) = 1 - p$$



Initial knowledge $\kappa = (10,2.5)$

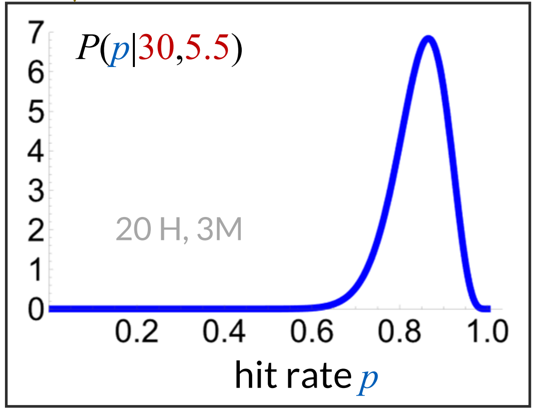
(via previous test event)



+ Data

111110111111
11110110111

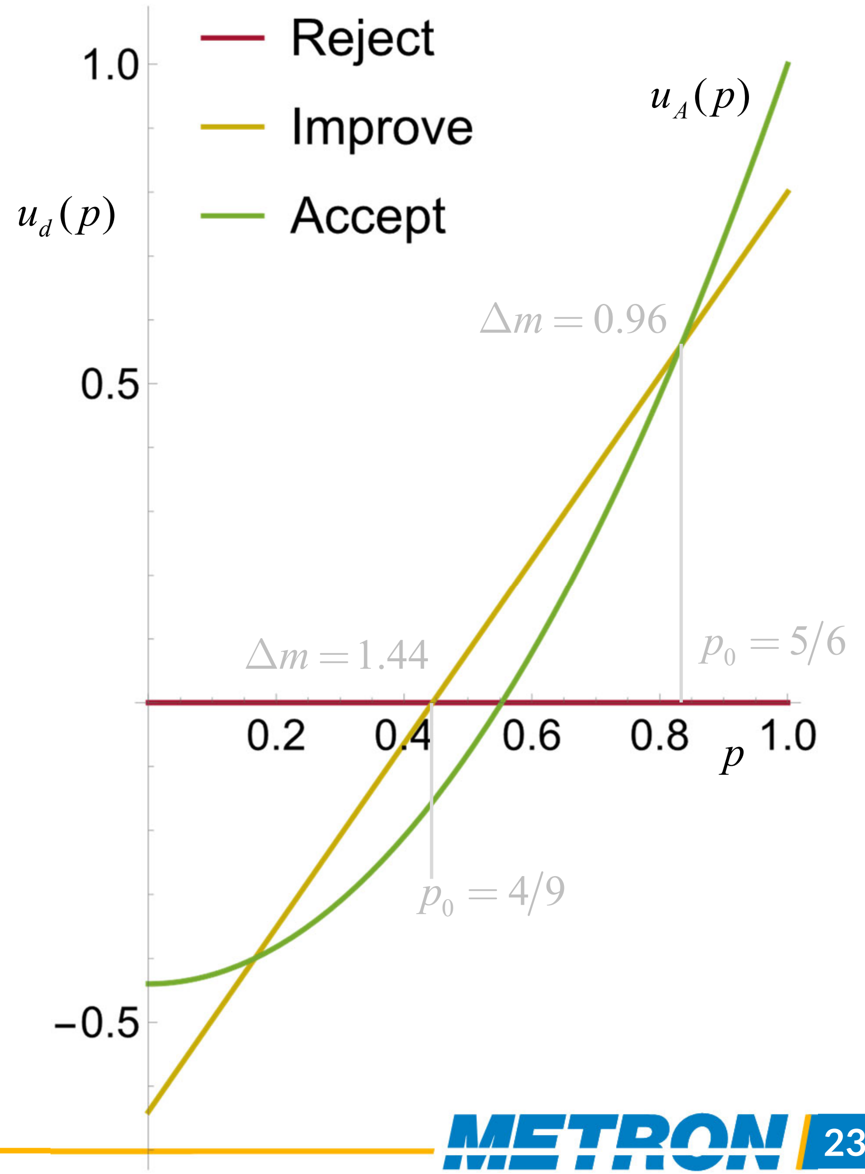
Updated knowledge $\kappa = (30,5.5)$



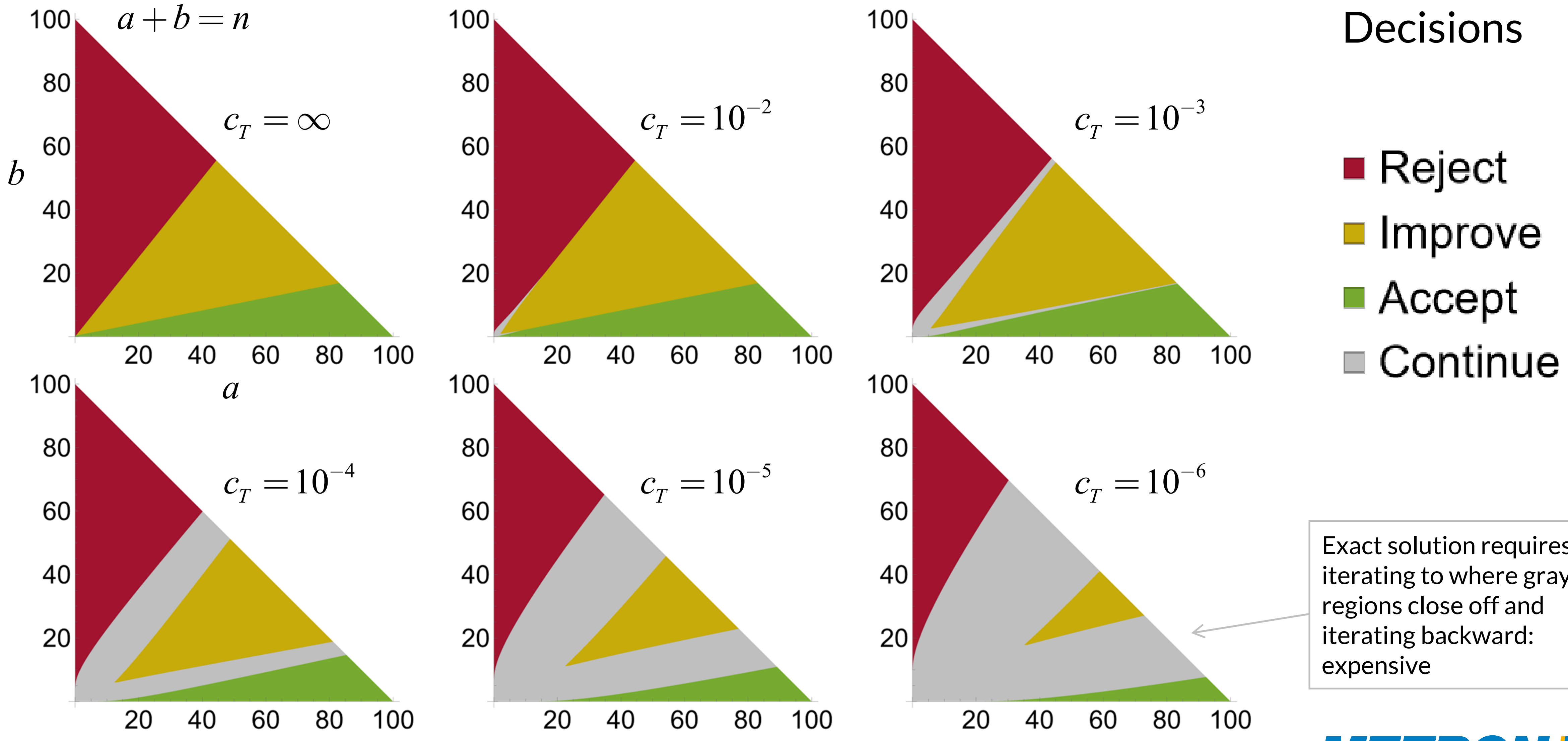
J. Ferry “Experimental design for operational utility,” *The ITEA Journal of Test and Evaluation*, **44**(3), 2023

Terminal Utilities for Hit/Miss System

- Specialize equations to simple hit/miss case
  - Terminal decisions: $D = \{R, I, A\}$
    - Reject, Improve, or Accept
  - Continue Testing decisions: $C = \{T\}$
    - Test
  - Intrinsic utilities
    - $u_R(p) = 0$
    - $u_I(p) = 1.44\,p - 0.64$
    - $u_A(p) = 1.44\,p^2 - 0.44$
  - Best to Reject for $p < 4/9$
  - Best to Improve for $4/9 < p < 5/6$
  - Best to Accept for $5/6 < p$

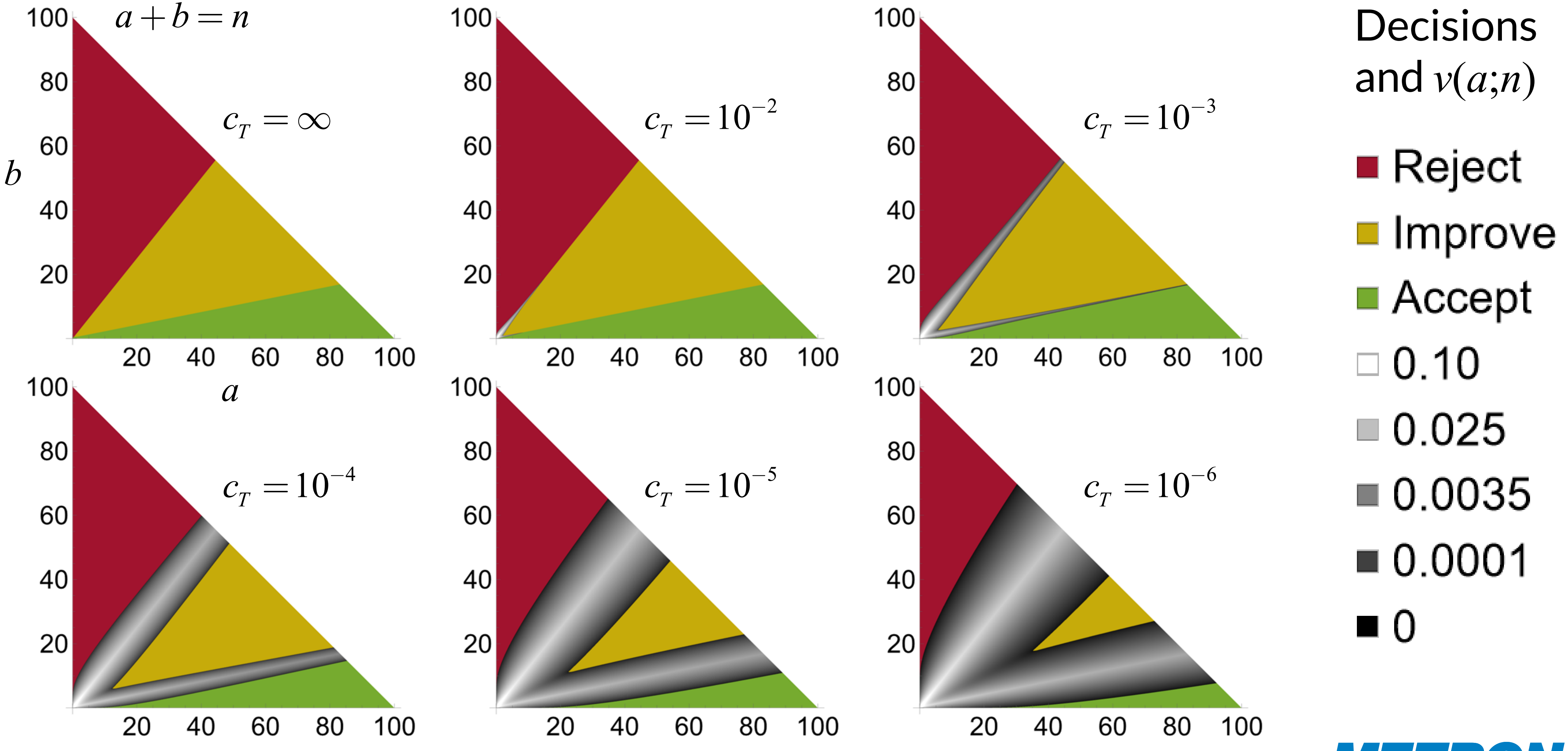


Optimal Decisions as a Function of Cost of Single Trial

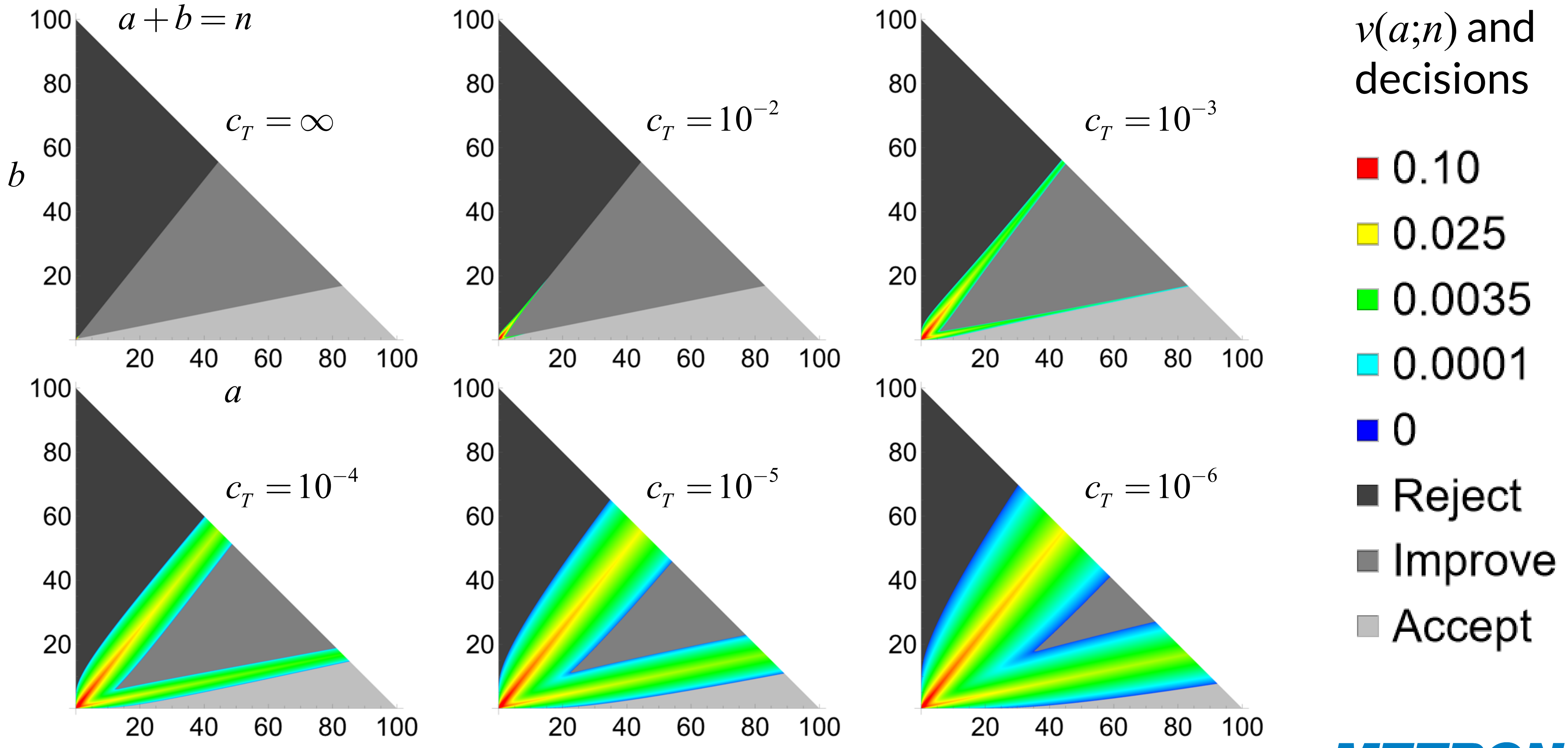


Exact solution requires iterating to where gray regions close off and iterating backward: expensive

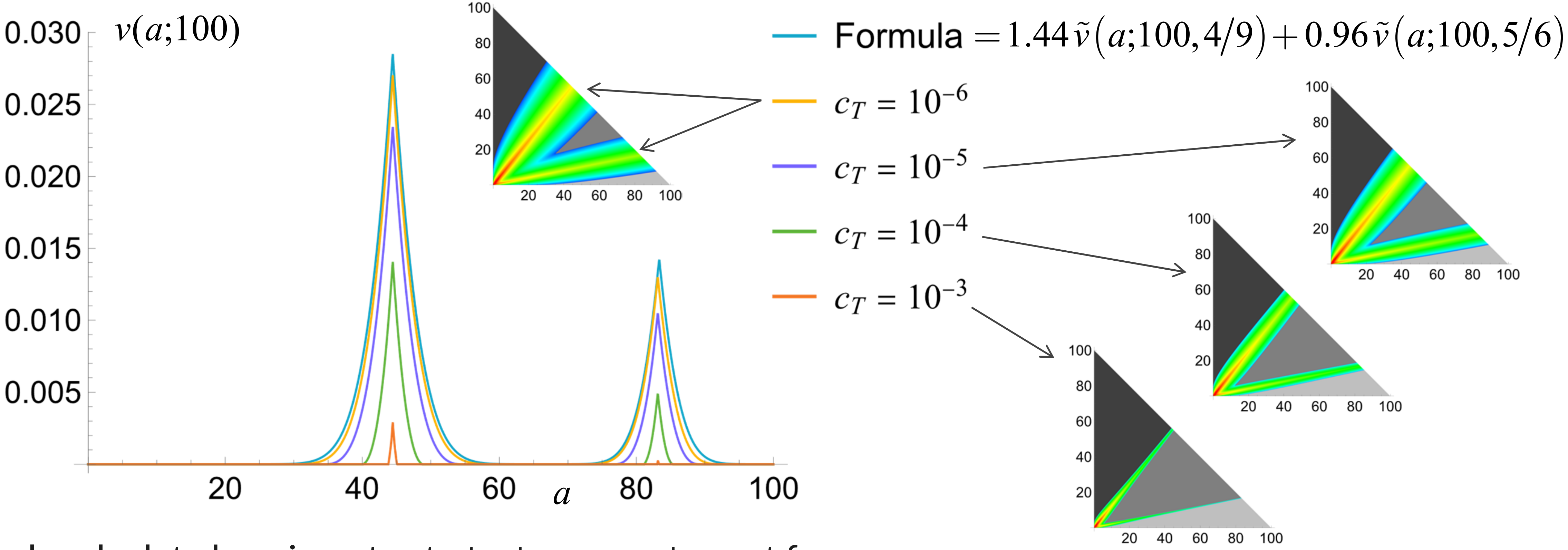
Shade Continue Region with Marginal Utility



Change Color Scheme to Focus on Marginal Utility



Solve for Marginal Utility *v(a;n)* Directly?



- Goal: calculate how impetus to test propagates out from decision boundaries – enables direct computation of optimal decisions
- Exact asymptotic formula for zero-cost case: $\tilde{v}(a;n,p_0) \doteq \sqrt{\dfrac{p_0(1-p_0)}{4n}}\,F\!\left(\dfrac{a-np_0}{\sqrt{np_0(1-p_0)}}\right)$ with $F(z) \doteq \dfrac{|z|}{2\sqrt{\pi}}\Gamma\!\left(-\dfrac{1}{2},\dfrac{z^2}{2}\right)$

Application of Dynamo to AN/TPQ-53 System

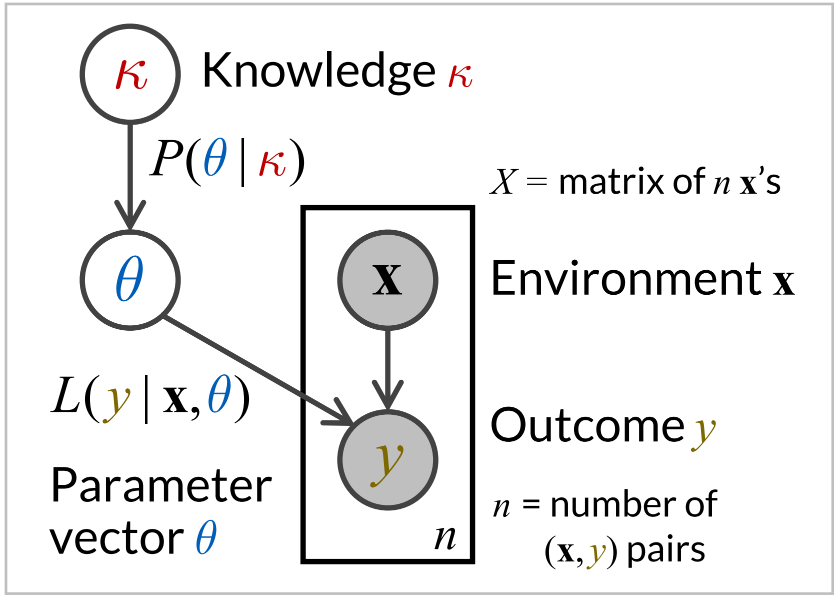
- Governing equations for Bayesian linear regression model

Normal–Normal-Inverse-Gamma (NNIG) Model

- Parameter $\theta = (\mathbf{c}, \sigma^2)$ governs distribution of y for each value of $\mathbf{x}$
  - Scalar version: $L(y \mid \mathbf{x}, \theta) = \mathcal{N}(y; \mathbf{c}\cdot\mathbf{x}, \sigma^2) \doteq \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - \mathbf{c}\cdot\mathbf{x})^2}{2\sigma^2} \right)$
  - Vector version: $L(\vec{y} \mid X, \theta) = \mathcal{N}(\vec{y}; X\mathbf{c}^T, \sigma^2 I) \doteq \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{(\vec{y} - X\mathbf{c}^T)^T (\vec{y} - X\mathbf{c}^T)}{2\sigma^2} \right)$
- Knowledge $\kappa = (\boldsymbol{\mu}, V, \alpha, \beta)$ governs distribution of θ
  - $P(\theta \mid \kappa) = P(\mathbf{c} \mid \boldsymbol{\mu}, V, \sigma^2) P(\sigma^2 \mid \alpha, \beta)$

$$P(\mathbf{c} \mid \boldsymbol{\mu}, V, \sigma^2) = \mathcal{N}(\mathbf{c}; \boldsymbol{\mu}, \sigma^2 V) \doteq \frac{1}{\sqrt{(2\pi\sigma^2)^d \, |V|}} \exp\left(-\frac{(\mathbf{c} - \boldsymbol{\mu}) V^{-1} (\mathbf{c} - \boldsymbol{\mu})^T}{2\sigma^2} \right)$$

$$P(\sigma^2 \mid \alpha, \beta) = \mathcal{IG}\left(\sigma^2; \frac{\alpha}{2}, \frac{\beta}{2} \right) = \frac{(\beta/2)^{\alpha/2}}{\Gamma(\alpha/2)} (\sigma^2)^{-(\alpha/2+1)} e^{-\beta/(2\sigma^2)}$$



Bayesian Inversion to Assimilate Test Data

- Posterior predictive distribution = multivariate *t*
  - $P(\vec{y} \mid X, \kappa) = \int L(\vec{y} \mid X, \theta) P(\theta \mid \kappa)\, d\theta = \mathcal{T}_{\alpha}\left(\vec{y}; X\boldsymbol{\mu}^T, \frac{\beta}{\alpha}\left(I + XVX^T\right)\right)$

$$= \frac{\Gamma\left((\alpha+n)/2\right)}{\Gamma(\alpha/2)\sqrt{(\beta\pi)^n \,|I + XVX^T|}}\left(1 + \frac{1}{\beta}\left(\vec{y} - X\boldsymbol{\mu}^T\right)^T \left(I + XVX^T\right)^{-1}\left(\vec{y} - X\boldsymbol{\mu}^T\right)\right)^{-(\alpha+n)/2}$$

Property #1:
predict outcomes y
given knowledge κ

- Bayesian inversion
  - $P(\theta \mid \vec{y}, X, \kappa_0) = \frac{L(\vec{y} \mid X, \theta) P(\theta \mid \kappa_0)}{P(\vec{y} \mid X, \kappa_0)} = \frac{\mathcal{N}(\vec{y}; X\mathbf{c}^T, \sigma^2 I)\mathcal{N}(\mathbf{c}; \boldsymbol{\mu}_0, \sigma^2 V_0)\mathcal{IG}(\sigma^2; \alpha_0/2, \beta_0/2)}{\mathcal{T}_{\alpha_0}\left(\vec{y}; X\boldsymbol{\mu}_0^T, \frac{\beta_0}{\alpha_0}\left(I + XV_0X^T\right)\right)}$
- Conjugate prior structure: simple updates
  - $P(\theta \mid \vec{y}, X, \kappa_0) = P(\theta \mid \kappa_n) = P(\mathbf{c} \mid \boldsymbol{\mu}_n, V_n, \sigma^2) P(\sigma^2 \mid \alpha_n, \beta_n)$

Property #2:
update knowledge κ
given outcomes y

$\boldsymbol{\mu}, \mathbf{c}, \mathbf{x} : 1 \times d$

$\vec{y} : n \times 1$

$V : d \times d$

$X : n \times d$

Conjugate Prior Structure of NNIG: Simple Updating

- Update rule for κ
  - $$\boldsymbol{\mu}_n = \left(\boldsymbol{\mu}_0 V_0^{-1} + \vec{y}^T X\right) V_n$$
    $$V_n = \left(V_0^{-1} + X^T X\right)^{-1}$$
    $$\alpha_n = \alpha_0 + n$$
    $$\beta_n = \beta_0 + \left(\vec{y} - X\boldsymbol{\mu}_0^T\right)^T \left(I + X V_0 X^T\right)^{-1} \left(\vec{y} - X\boldsymbol{\mu}_0^T\right)$$
- Simplified update: let $W \doteq V^{-1}$, $\mathbf{v} \doteq \boldsymbol{\mu} W$, and $\gamma \doteq \beta + \boldsymbol{\mu} \cdot \mathbf{v}$
  - $$\mathbf{v}_n = \mathbf{v}_0 + \vec{y}^T X$$
    $$W_n = W_0 + X^T X$$
    $$\alpha_n = \alpha_0 + n$$
    $$\gamma_n = \gamma_0 + \vec{y}^T \vec{y}$$

$\boldsymbol{\mu}, \mathbf{v}, \mathbf{c}, \mathbf{x} : 1 \times d$

$\vec{y} : n \times 1$

$V, W : d \times d$

$X : n \times d$

Example of Knowledge Updating

- Bayesian model maintains the knowledge $\kappa = (\boldsymbol{\mu}, V, \alpha, \beta)$ about system
  - Beginning with Subject Matter Expert (SME) knowledge initially
    - Though this example uses a diffuse prior (no initial knowledge)
  - Knowledge updated with each test
    - Each test is a pair $(\mathbf{x}, y)$: an environment $\mathbf{x}$ and an outcome y
- Sequential Bayesian Testing: can use κ to assess system at any time
- Four-slide example for proxy data
  - Knowledge κ: determines distribution over parameter θ
  - Any value of the parameter θ predicts outcomes y in any environment $\mathbf{x}$
    - Outcome y = log(error) between actual and estimated location of an object
    - Environment $\mathbf{x}$ = various discrete and continuum factors that influence outcome
  - Knowledge κ updated after every test result $(\mathbf{x}, y)$

Sequential Bayesian Updates with TPQ-53 Proxy Data

Knowledge κ

$$\bar{\sigma} \doteq \sqrt{\frac{\beta}{\alpha - 2}}$$

10 α

0.8 $\bar{\sigma}$

μ

$$\begin{pmatrix} 0.81 & -0.41 & 0.57 & -0.16 & -0.46 & 0.46 & 0.23 & -0.23 & 0.14 & 0.03 & 0.35 \end{pmatrix}$$

V

$$\begin{pmatrix}
643 & 178 & -250 & 71 & 203 & -203 & -100 & 100 & -62 & -12 & -153 \\
178 & 577 & -208 & -369 & -102 & 102 & 50 & -50 & 31 & 6 & 77 \\
-250 & -208 & 492 & -283 & 142 & -142 & -70 & 70 & -43 & -8 & -107 \\
71 & -369 & -283 & 652 & -41 & 41 & 20 & -20 & 12 & 2 & 31 \\
203 & -102 & 142 & -41 & 384 & -384 & 57 & -57 & 35 & 7 & 87 \\
-203 & 102 & -142 & 41 & -384 & 384 & -57 & 57 & -35 & -7 & -87 \\
-100 & 50 & -70 & 20 & 57 & -57 & 472 & -472 & -17 & -3 & -43 \\
100 & -50 & 70 & -20 & -57 & 57 & -472 & 472 & 17 & 3 & 43 \\
-62 & 31 & -43 & 12 & 35 & -35 & -17 & 17 & 989 & -2 & -27 \\
-12 & 6 & -8 & 2 & 7 & -7 & -3 & 3 & -2 & 1000 & -5 \\
-153 & 77 & -107 & 31 & 87 & -87 & -43 & 43 & -27 & -5 & 934
\end{pmatrix}$$

Example parameters θ

0.72 σ

$\mathbf{c}$

$$\begin{pmatrix} -28.32 & 2.2 & 47.9 & -50.1 & -0.1 & 0.1 & 8.13 & -8.13 & -12.51 & 1.24 & -32.95 \end{pmatrix}$$

0.84

$$\begin{pmatrix} -6.96 & 2.16 & 1.7 & -3.86 & 19.69 & -19.69 & 32.95 & -32.95 & -12.53 & -23.94 & 36.01 \end{pmatrix}$$

- Knowledge κ after 10 tests
  - Determines probabilities for parameters θ
- Two examples of parameters θ
  - Determines probabilities for outcomes y in any environment $\mathbf{x}$
- Four examples of y for each θ

Environment $\mathbf{x}$

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 8.97935 & 0.294524 & 9.4727 \end{pmatrix}$$

Example errors y (in meters)

$$\begin{pmatrix} 5.2\times10^{-52} & 8.1\times10^{-52} & 1.8\times10^{-51} & 4.8\times10^{-52} \\ 4.7\times10^{10} & 2.1\times10^{10} & 3.2\times10^{10} & 6.8\times10^{10} \end{pmatrix}$$

Sequential Bayesian Updates with TPQ-53 Proxy Data

Knowledge κ

50

0.59

$$(1.68 \;\; -0.22 \;\; 0.56 \;\; -0.34 \;\; 0.09 \;\; -0.09 \;\; 0.47 \;\; -0.47 \;\; 0.84 \;\; -0.52 \;\; -0.62)$$

178.4	-81.9	-82.4	164.3	0.	0.	-230.	230.	-0.3	-0.3	0.1
-81.9	158.8	158.2	-316.9	-0.1	0.1	-22.9	22.9	-0.4	-0.4	0.1
-82.4	158.2	158.7	-316.9	0.1	-0.1	-23.1	23.1	0.3	0.3	-0.1
164.3	-316.9	-316.9	633.8	0.	0.	46.	-46.	0.1	0.1	0.
0.	-0.1	0.1	0.	0.2	-0.2	0.	0.	0.2	-0.1	0.2
0.	0.1	-0.1	0.	-0.2	0.2	0.	0.	-0.2	0.1	-0.2
-230.	-22.9	-23.1	46.	0.	0.	435.6	-435.6	-0.1	-0.1	0.
230.	22.9	23.1	-46.	0.	0.	-435.6	435.6	0.1	0.1	0.
-0.3	-0.4	0.3	0.1	0.2	-0.2	-0.1	0.1	0.6	0.4	0.
-0.3	-0.4	0.3	0.1	-0.1	0.1	-0.1	0.1	0.4	1.1	-0.1
0.1	0.1	-0.1	0.	0.2	-0.2	0.	0.	0.	-0.1	0.4

- Knowledge κ after 50 tests
  - I.e., updated using 50 pairs $(\mathbf{x}, y)$
- Predicted outcomes y still way off
  - Due to initial κ set to "no knowledge"
  - In practice, SME knowledge provides initial κ that yields reasonable y values

Example parameters θ

0.56

$$(-1.65 \;\; 1.23 \;\; 1.86 \;\; -3.1 \;\; -0.01 \;\; 0.01 \;\; 5.06 \;\; -5.06 \;\; 0.73 \;\; -0.32 \;\; -0.6)$$

0.63

$$(5.3 \;\; -6.15 \;\; -5.34 \;\; 11.49 \;\; 0.1 \;\; -0.1 \;\; 0.73 \;\; -0.73 \;\; 0.52 \;\; -0.81 \;\; -0.82)$$

Environment $\mathbf{x}$

$$(1 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 8.97935 \;\; 0.294524 \;\; 9.4727)$$

Example errors y (in meters)

$$\begin{pmatrix} 0.049 & 0.042 & 0.098 & 0.069 \\ 3.3\times10^{8} & 1.4\times10^{8} & 5.9\times10^{8} & 1.3\times10^{8} \end{pmatrix}$$

Sequential Bayesian Updates with TPQ-53 Proxy Data

Knowledge κ

150

0.65

$(2.22\ \ -0.17\ \ -0.24\ \ 0.41\ \ -0.1\ \ 0.1\ \ 0.62\ \ -0.62\ \ 0.44\ \ -0.94\ \ -0.62)$

135.62	0.03	-0.04	0.02	-0.01	0.01	-242.03	242.03	-0.09	-0.02	0.01
0.03	0.07	-0.04	-0.03	-0.01	0.01	0.01	-0.01	-0.04	-0.05	0.05
-0.04	-0.04	0.06	-0.02	0.02	-0.02	-0.01	0.01	0.06	0.02	-0.03
0.02	-0.03	-0.02	0.05	-0.01	0.01	0.01	-0.01	-0.02	0.03	-0.02
-0.01	-0.01	0.02	-0.01	0.02	-0.02	0.	0.	0.03	0.01	0.01
0.01	0.01	-0.02	0.01	-0.02	0.02	0.	0.	-0.03	-0.01	-0.01
-242.03	0.01	-0.01	0.01	0.	0.	432.23	-432.23	-0.02	-0.01	0.
242.03	-0.01	0.01	-0.01	0.	0.	-432.23	432.23	0.02	0.01	0.
-0.09	-0.04	0.06	-0.02	0.03	-0.03	-0.02	0.02	0.12	0.04	-0.01
-0.02	-0.05	0.02	0.03	0.01	-0.01	-0.01	0.01	0.04	0.2	0.02
0.01	0.05	-0.03	-0.02	0.01	-0.01	0.	0.	-0.01	0.02	0.1

- Knowledge κ after 150 tests
  - For environment $\mathbf{x}$
  - Now produces reasonable y values
- Still some high uncertainty in κ
  - First 150 tests did not include diverse range of environments

Example parameters θ

0.6

$(6.16\ \ -0.06\ \ -0.36\ \ 0.41\ \ 0.02\ \ -0.02\ \ -6.56\ \ 6.56\ \ 0.57\ \ -0.97\ \ -0.3)$

0.6

$(10.89\ \ -0.24\ \ -0.33\ \ 0.57\ \ -0.17\ \ 0.17\ \ -15.\ \ 15.\ \ 0.52\ \ -1.23\ \ -0.8)$

Environment $\mathbf{x}$

Rocket, SR 90°

$(1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1\ \ 0\ \ 8.97935\ \ 0.294524\ \ 9.4727)$

Example errors y (in meters)

$$\begin{pmatrix} 26.2 & 22.9 & 79.2 & 73. \\ 36.1 & 49.2 & 41.2 & 28.3 \end{pmatrix}$$

Sequential Bayesian Updates with TPQ-53 Proxy Data

Knowledge κ

0.6

$$(2.5 \;\; -0.07 \;\; -0.16 \;\; 0.23 \;\; -0.02 \;\; 0.02 \;\; -0.28 \;\; 0.28 \;\; 0.68 \;\; -0.93 \;\; -0.26)$$

0.006	0.002	-0.001	-0.001	0.001	-0.001	0.001	-0.001	-0.005	0.	0.003
0.002	0.012	-0.004	-0.008	0.	0.	0.002	-0.002	-0.002	-0.006	0.011
-0.001	-0.004	0.005	-0.001	0.	0.	0.	0.	0.002	0.	-0.004
-0.001	-0.008	-0.001	0.008	-0.001	0.001	-0.001	0.001	0.001	0.006	-0.007
0.001	0.	0.	-0.001	0.002	-0.002	0.001	-0.001	-0.001	-0.001	0.
-0.001	0.	0.	0.001	-0.002	0.002	-0.001	0.001	0.001	0.001	0.
0.001	0.002	0.	-0.001	0.001	-0.001	0.003	-0.003	-0.001	-0.001	0.002
-0.001	-0.002	0.	0.001	-0.001	0.001	-0.003	0.003	0.001	0.001	-0.002
-0.005	-0.002	0.002	0.001	-0.001	0.001	-0.001	0.001	0.006	0.	-0.004
0.	-0.006	0.	0.006	-0.001	0.001	-0.001	0.001	0.	0.037	0.004
0.003	0.011	-0.004	-0.007	0.	0.	0.002	-0.002	-0.004	0.004	0.017

- Knowledge κ after all tests
- Example parameters θ
  - Clustered around mean
  - With low uncertainty
- Were all tests needed?

Example parameters θ

0.59

$$(2.53 \;\; -0.13 \;\; -0.2 \;\; 0.32 \;\; -0.03 \;\; 0.03 \;\; -0.24 \;\; 0.24 \;\; 0.58 \;\; -0.96 \;\; -0.28)$$

0.58

$$(2.51 \;\; -0.1 \;\; -0.16 \;\; 0.26 \;\; 0.01 \;\; -0.01 \;\; -0.27 \;\; 0.27 \;\; 0.64 \;\; -0.77 \;\; -0.25)$$

Environment $\mathbf{x}$

$$(1 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 8.97935 \;\; 0.294524 \;\; 9.4727)$$

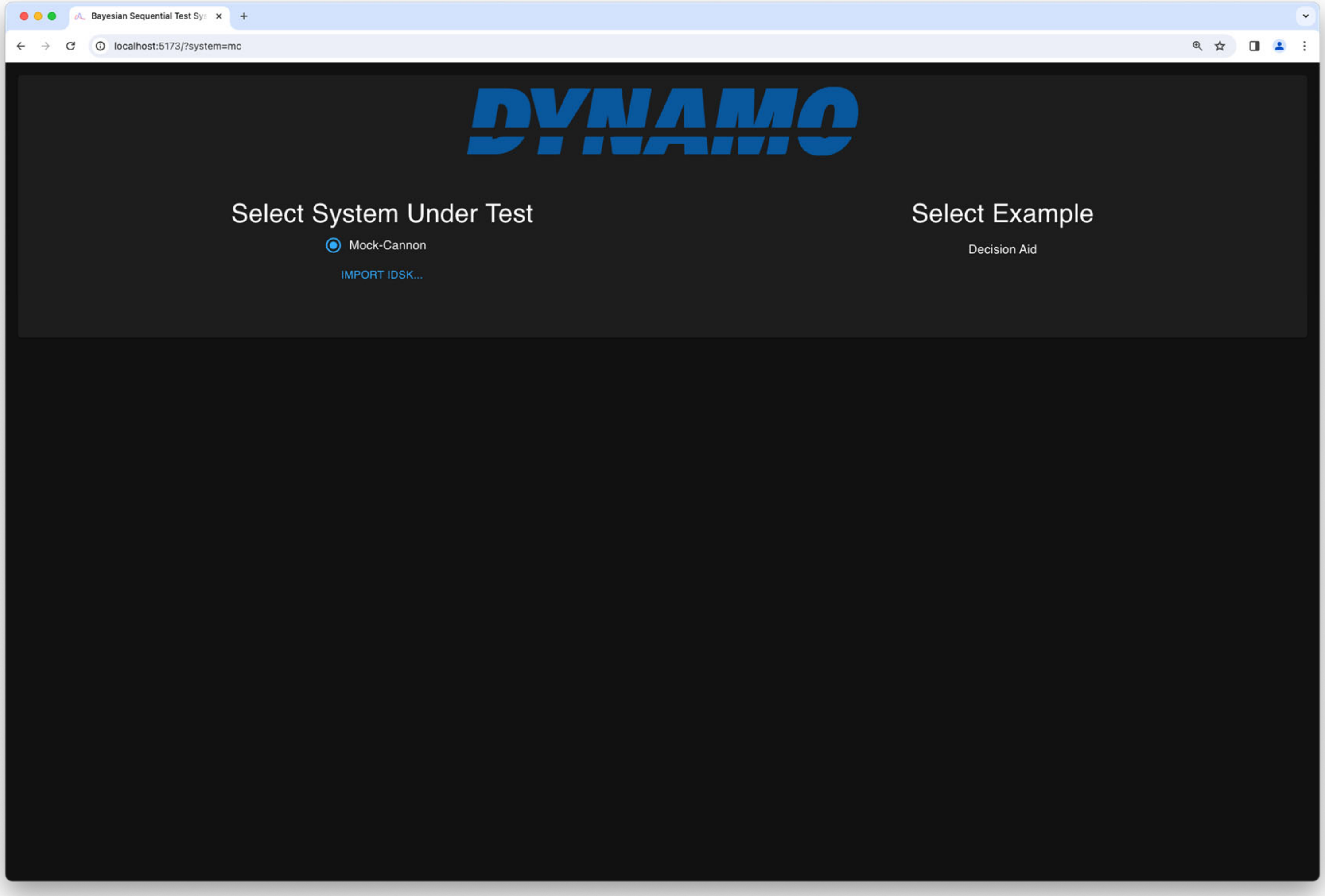
Example errors y (in meters)

$$\begin{pmatrix} 19. & 58.5 & 68.6 & 41.3 \\ 53.1 & 26. & 32.7 & 20.4 \end{pmatrix}$$

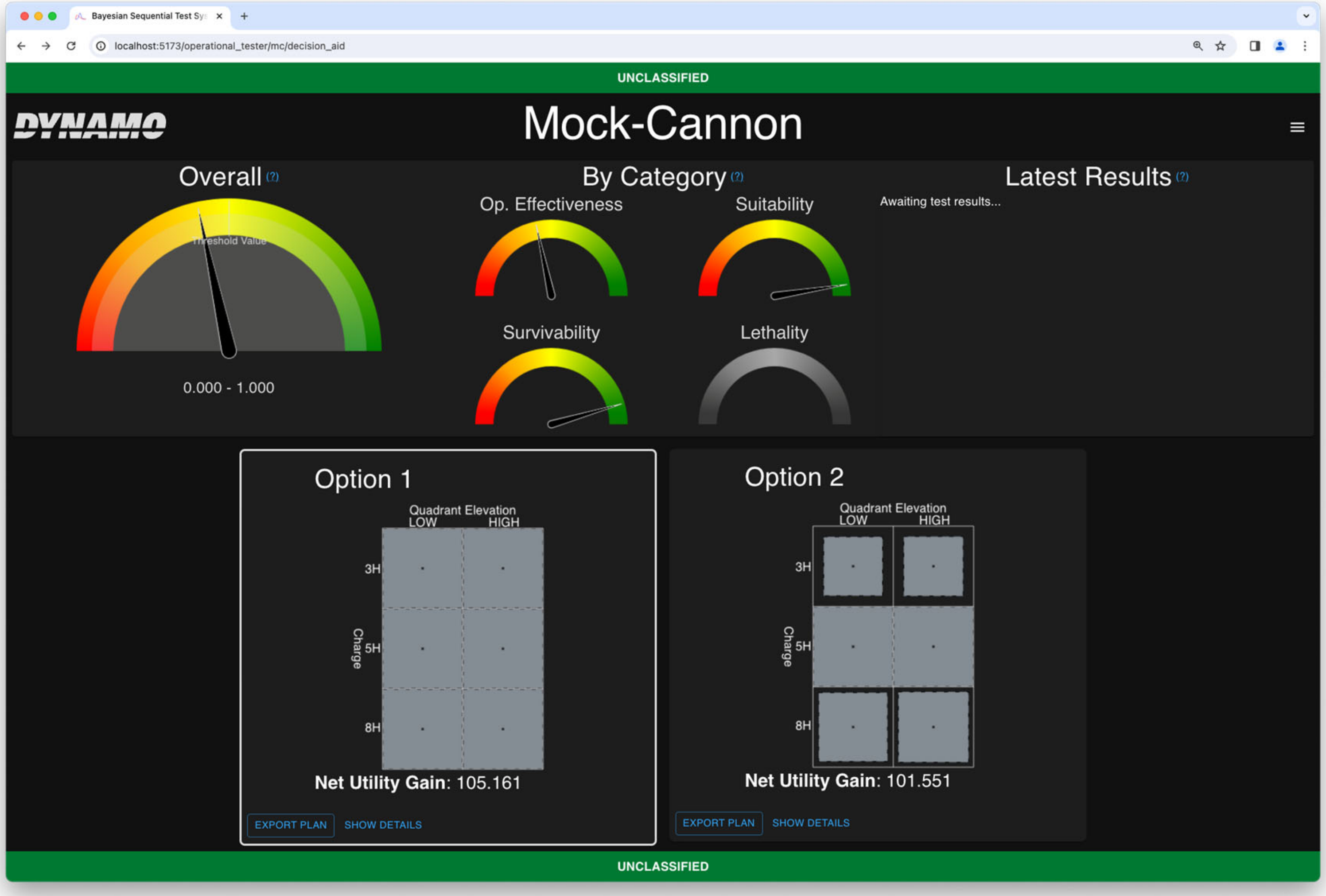
Preview of Forthcoming Dynamo GUI

- Unclassified Mock-Cannon example

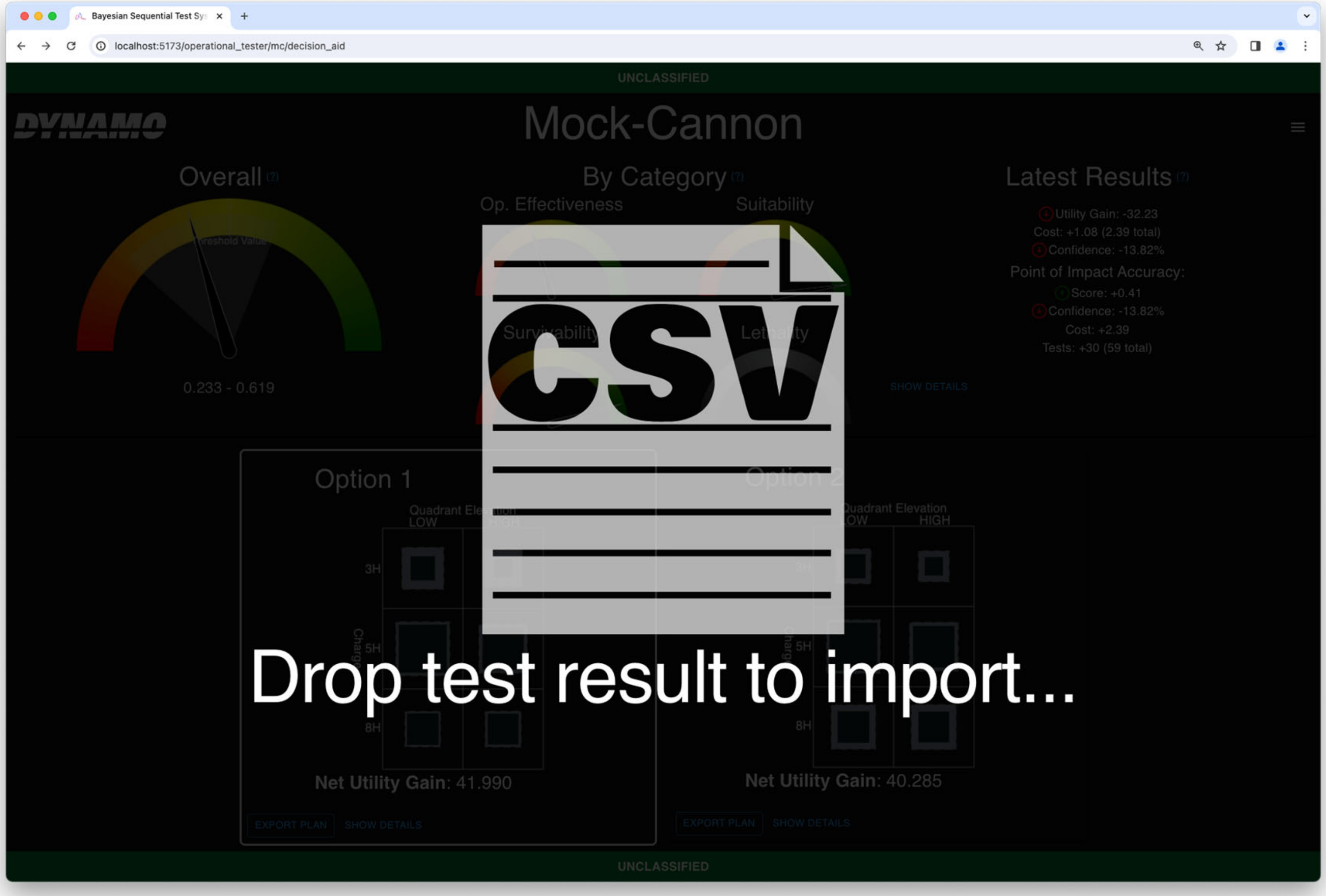
Mock-Cannon Example



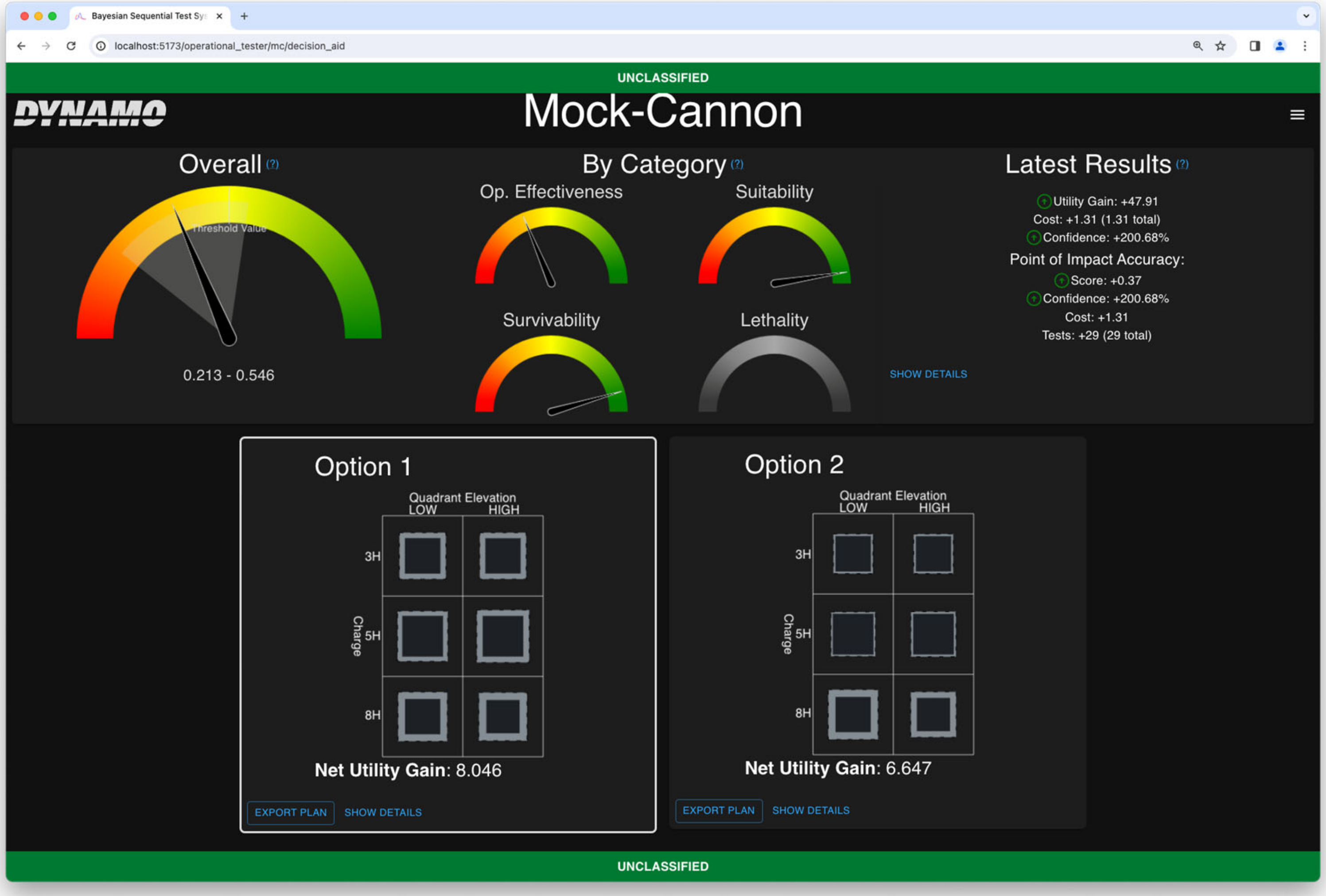
Mock-Cannon Example



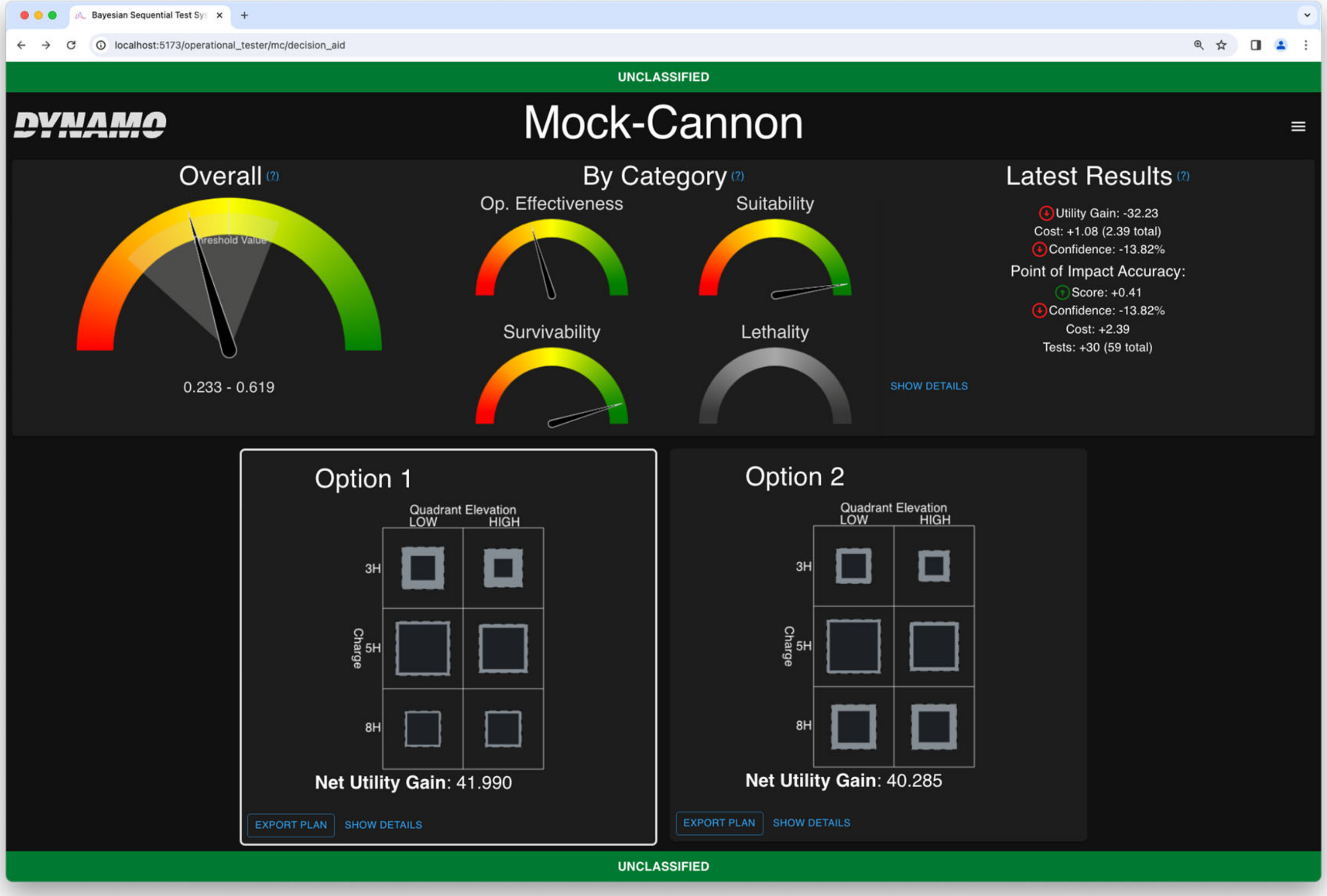
Mock-Cannon Example



Mock-Cannon Example



Mock-Cannon Example



Executive Summary

- Bayesian Sequential Testing
  - Bayesian model maintains *knowledge* about system under test
    - Enables knowledge to be ported between test events
    - Predicts impact of even a single trial on knowledge about system
  - Leverages multiple evaluation criteria
    - Requirements: visualize progress toward meeting requirements
    - Optimal design: generate test design candidates
    - Moneyball: novel criterion for Bayesian models
- Moneyball evaluation criterion
  - Based on operational utility of system given current knowledge
    - Captures stakeholder priorities
    - Formulated in same units as testing cost: *enables principled cost/benefit analysis*
  - Recommends which trials are best, or whether it's time to stop testing