UPDATES TO THE ANALOGY SOFTWARE COSTING TOOL (ASCOT) AND THE ONLINE NASA SPACE ESTIMATION TOOLS (ONSET)

Samuel R. Fleischer[1*]
Jairus M. Hihn[1]
James K. Johnson[2]

[1]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA

[2]National Aeronautics and Space Administration, Washington, DC

NASA Cost and Schedule Symposium
April 26-28, 2022, Cocoa Beach, Florida

*Corresponding author (Samuel.R.Fleischer@jpl.nasa.gov)

Jet Propulsion Laboratory
California Institute of Technology

Agenda

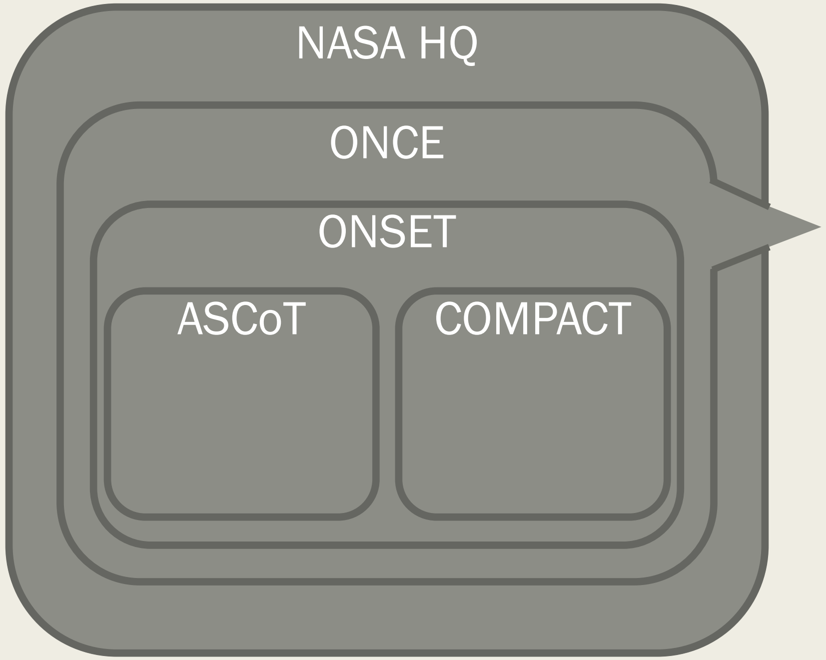
- ASCoT/ONSET Motivation
- ONSET Architecture
- ASCoT Toolset (See COMPACT Toolset at Melissa Hooke’s talk in about an hour)
- Parametric Models
  - *COCOMO (not covering today), Bayesian Regression*
- Analogic Models
  - *NLPCA, kNN, Clustering*
- The Future of ONSET

ASCoT/ONSET Motivation

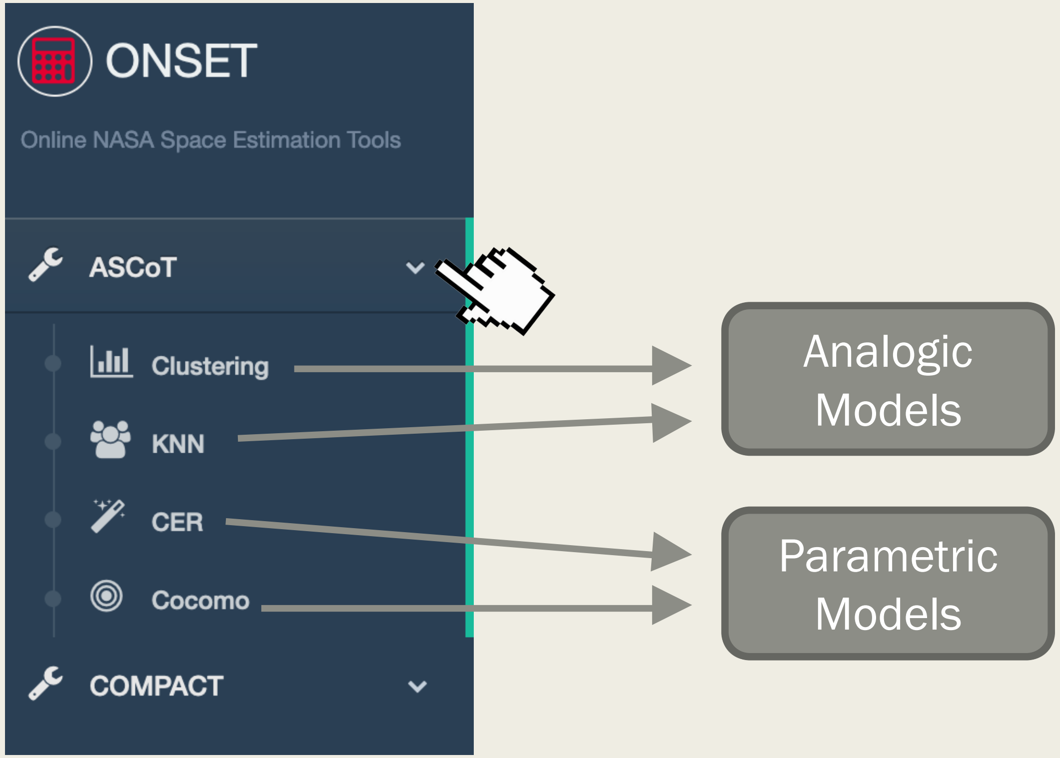
- More proposals = more concept design and trade studies
- More designs need to be evaluated quickly and early in the lifecycle with cost models that
  - *Don't require experts*
  - *Are transparent*
  - *Are easily accessible*
- ONSET was originally just ASCoT (software cost estimation) and has expanded to include COMPACT (CubeSat cost estimation)
- ONSET is an evolving suite of web-based tools expanding our costing tool set
- ONSET is a platform for <u>*your*</u> web-based costing tool!
  - *It's surprisingly simple to join – ask us how!*

ONSET Architecture

- Secure, Django framework
  - *ASCoT and COMPACT are Django "applications" built entirely using Dash*
- Surprisingly simple to make your own independent tool and plug it in to ONSET
- Hosted by NASA HQ on ONCE (One NASA Cost Engineering)
  - *Available to any NASA civil servant or contractor who has access to ONCE*



ASCoT Toolset

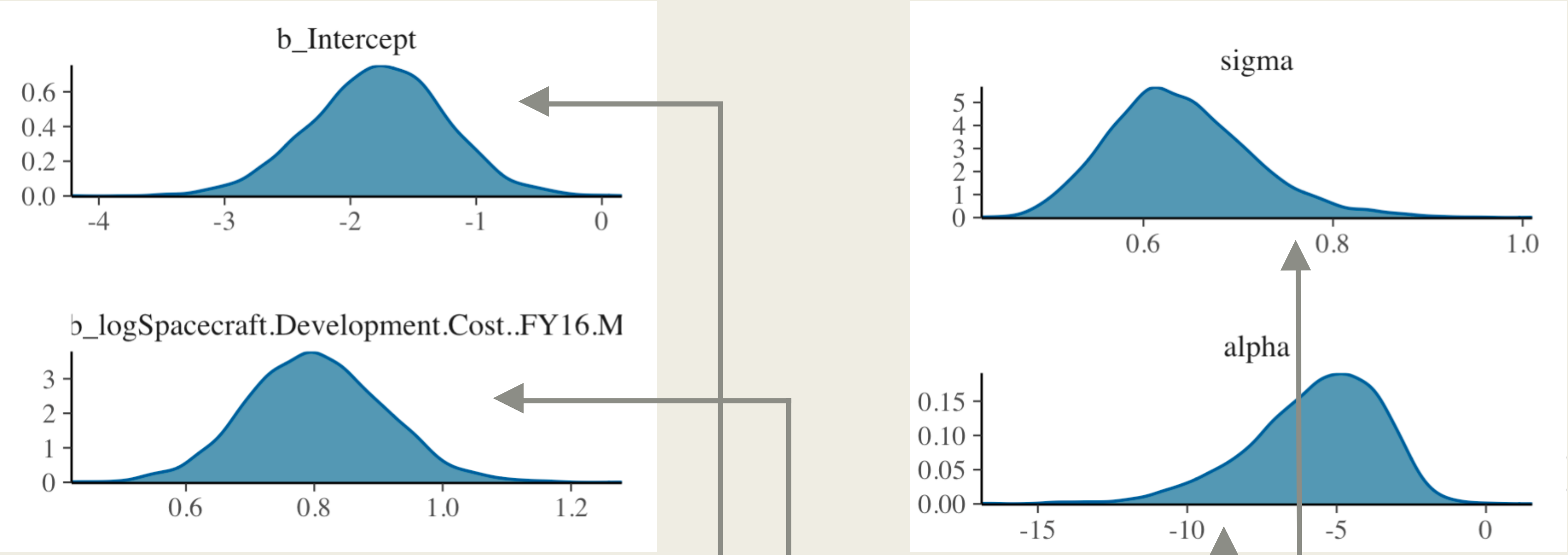


The data:

- N = 54
- Sources
  - NASA CADRe
  - JPL SMART repo
  - project documentation
  - direct interviews
  - Independent, industry-wide dataset
- Variables
  - Destination
  - Redundancy
  - Software inheritance
  - Mission type
  - Mission size
  - Number of instruments
  - Number of deployables
  - Cost
  - Effort

Jet Propulsion Laboratory
California Institute of Technology

Bayesian CER



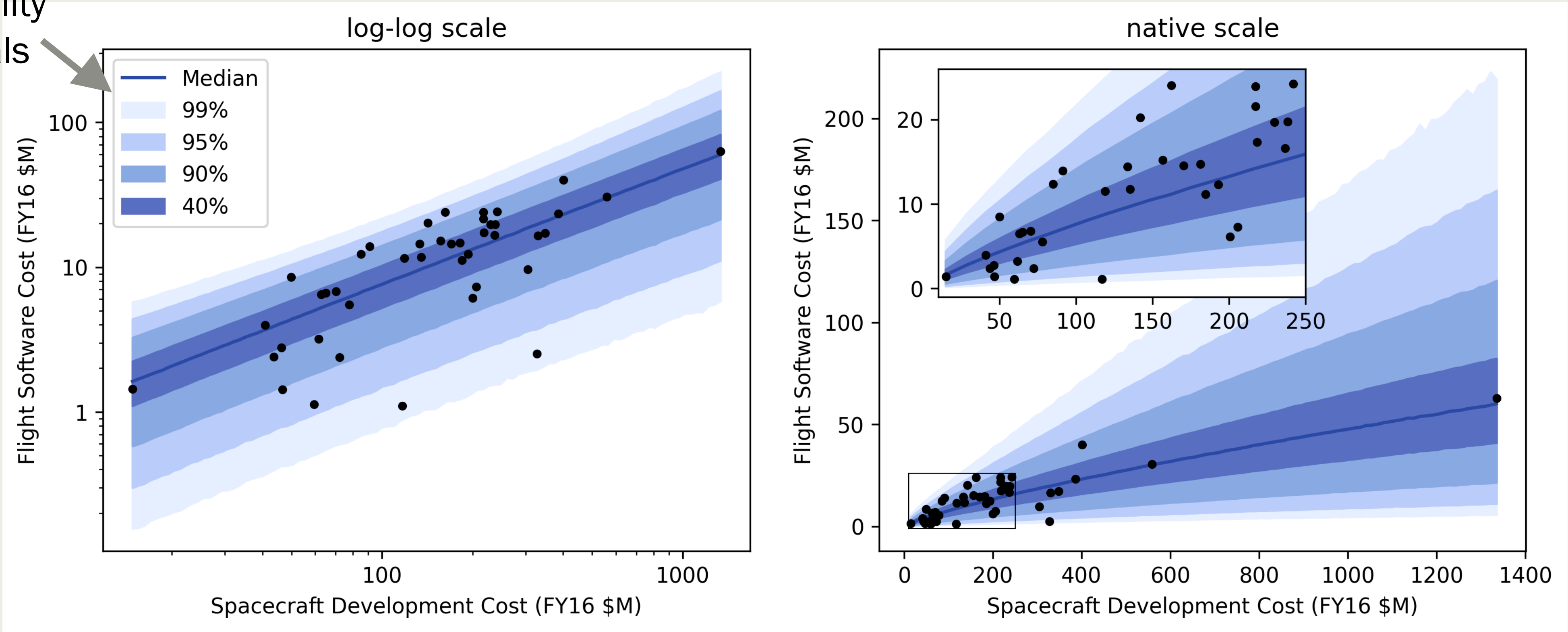
$$\log(\text{Software Cost}) = \beta_0 + \beta_1 \log(\text{Spacecraft Cost}) + \epsilon$$
$$\epsilon \sim \text{SkewNormal}(\sigma, \alpha)$$

Priors
$\alpha \sim N(0,4)$
$\sigma \sim t(3,0,2.5)$
$\beta_0 \sim t(3,2.5,2.5)$
$\beta_1 \sim U(-\infty, \infty)$

Jet Propulsion Laboratory
California Institute of Technology

Bayesian CER

credibility intervals

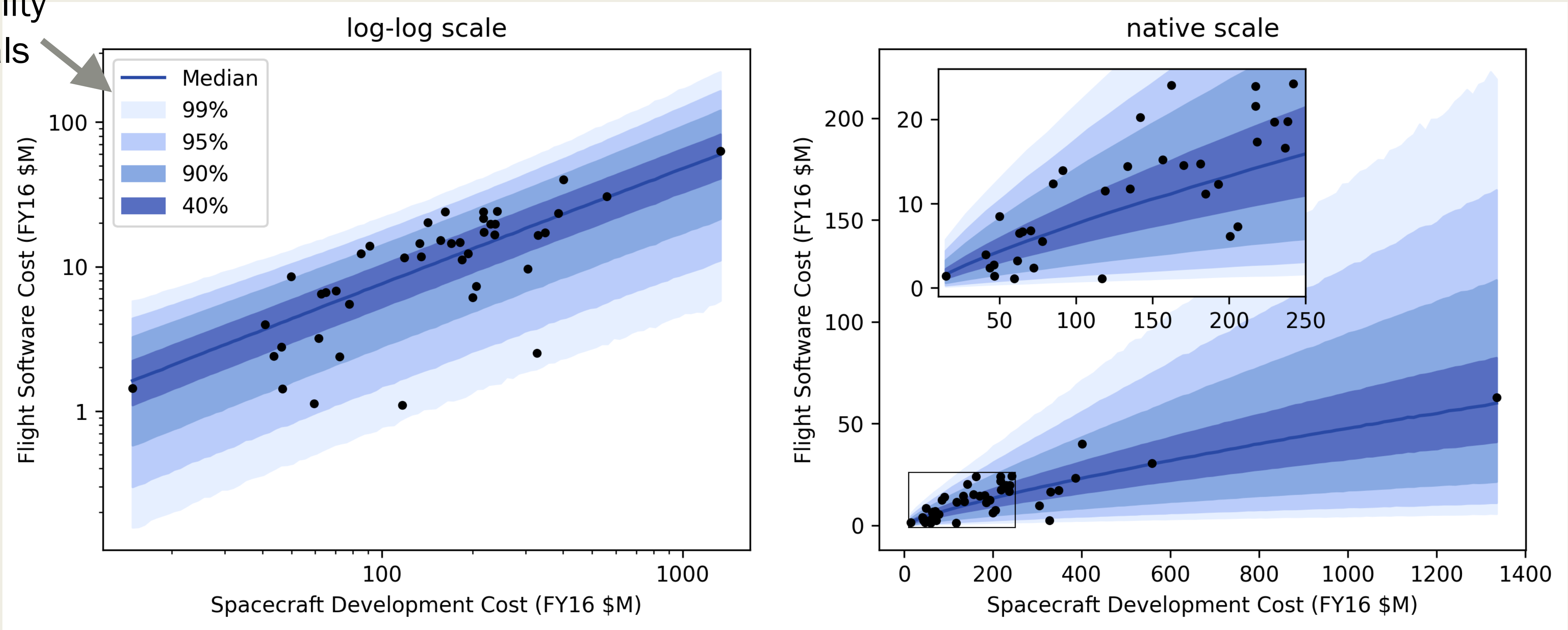


$$\log(\text{Software Cost}) = \beta_0 + \beta_1 \log(\text{Spacecraft Cost}) + \epsilon$$
$$\epsilon \sim \text{SkewNormal}(\sigma, \alpha)$$

Priors
$\alpha \sim N(0,4)$
$\sigma \sim t(3,0,2.5)$
$\beta_0 \sim t(3,2.5,2.5)$
$\beta_1 \sim U(-\infty, \infty)$

Bayesian CER

credibility intervals



- Skew normal error term performs better than normal error (log-skew-normal vs log-normal)
  - Captures low outliers without pulling median prediction down
  - Reasonable uncertainty bounds on the native scale due to negative skew
- Simple regression performs better than models including other software cost drivers such as number of instruments, destination, or redundancy (short version: avoids overfitting)

Analogic Models

- *k*NN model
  - *Finds the three missions most similar to your input*
  - *Estimate is a weighted average of these nearest neighbors*
- Cluster model
  - *Finds the mission cluster most suited to your input*
  - *Estimate is a weighted average of missions in your cluster*
- Both models utilize nonlinear principal components analysis (NLPCA)

NLPCA motivation

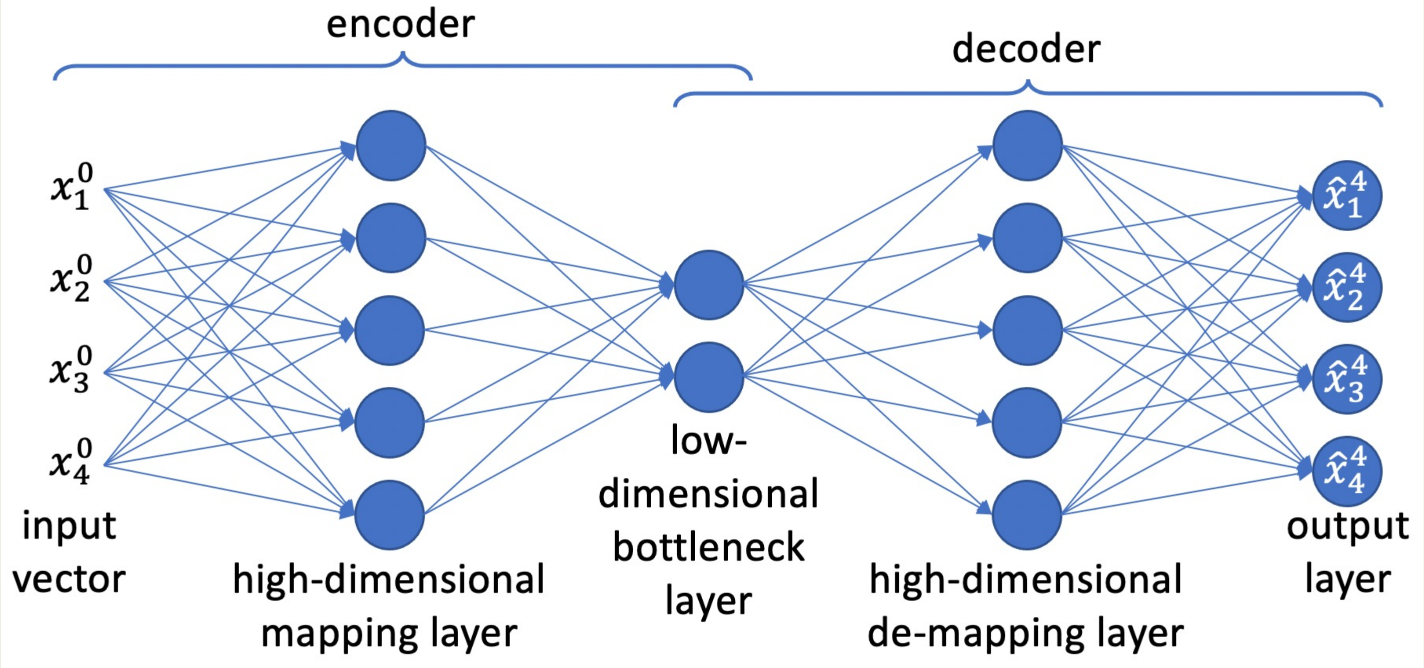
- How do we determine proximity of data when the data is numeric?
  - *Use a distance formula (Pythagorean or other)*
  - *Example*
    - Mission 1: (4 instruments, 5 deployables)
    - Mission 2: (2 instruments, 1 deployable)
    - Distance: $d = \sqrt{(4-2)^2 + (5-1)^2} = \sqrt{20}$
- How do we determine proximity of data when the data is NOT numeric?
  - *Example*
    - Mission 1: (Mars-bound, dual-string cold backup)
    - Mission 2: (Saturn-bound, dual-string warm backup)
    - Distance: $d = ???$

NLPCA motivation

- We have to find a way to numericize the data.
  - *Previous ASCoT versions chose "intuitive" transformations.*
  - *i.e. Single-string = 1, Dual-string (cold) = 2, Dual-string (warm) = 4.*
  - *This encapsulates the industry knowledge that the difference between a dual-string (warm) system and a dual-string (cold) system is greater than the difference between a dual-string (cold) system and a single-string system.*
- NLPCA lets the data speak for itself – optimal transformations are learned using machine learning… in particular auto-associative neural networks

NLPCA - ANNs

Auto-associative neural network



ANN parameters are optimized such that the difference between the output layer and the input layer is minimized.

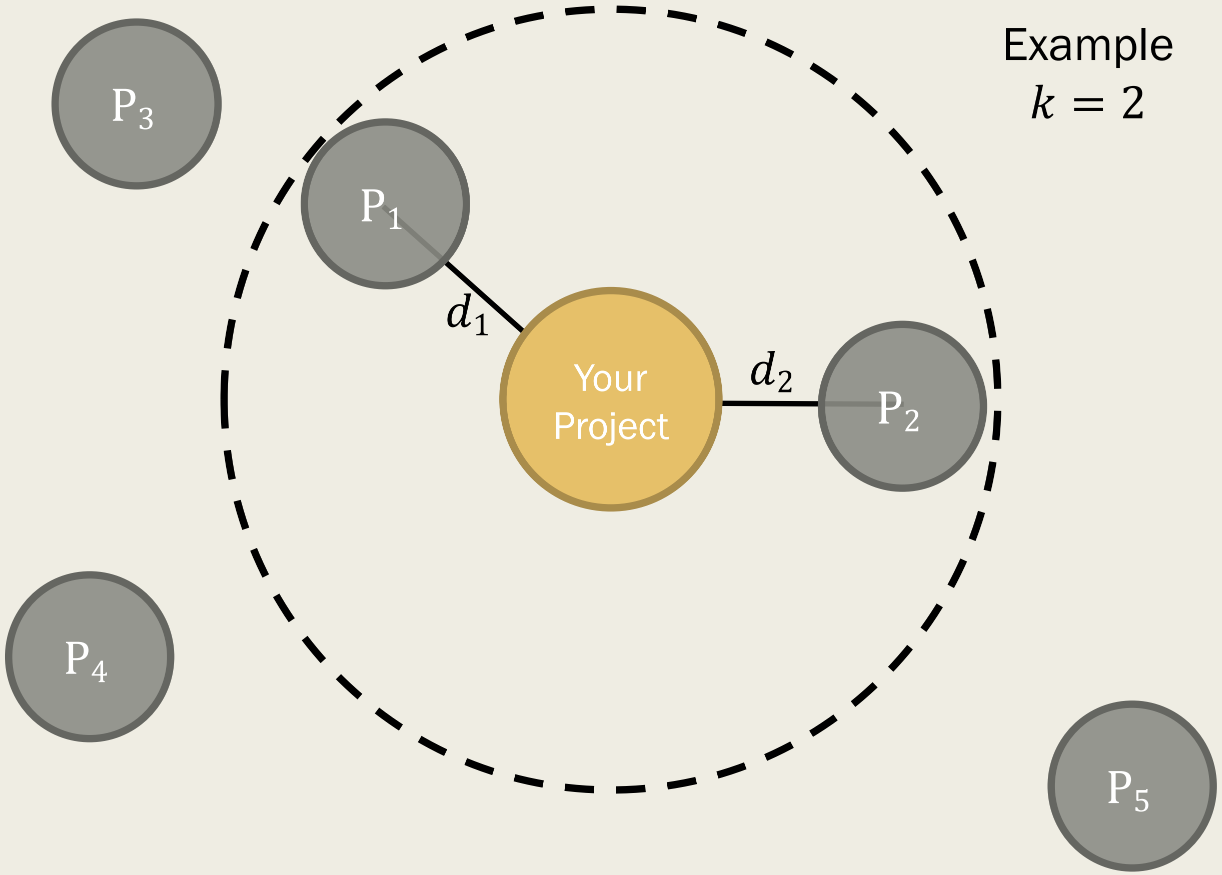
The goal is for the low-dimensional bottleneck layer to adequately retain the information contained in the input layer.

The result is that a non-numeric input layer can be projected onto a numeric, low-dimensional space.

*k*NN Algorithm Overview

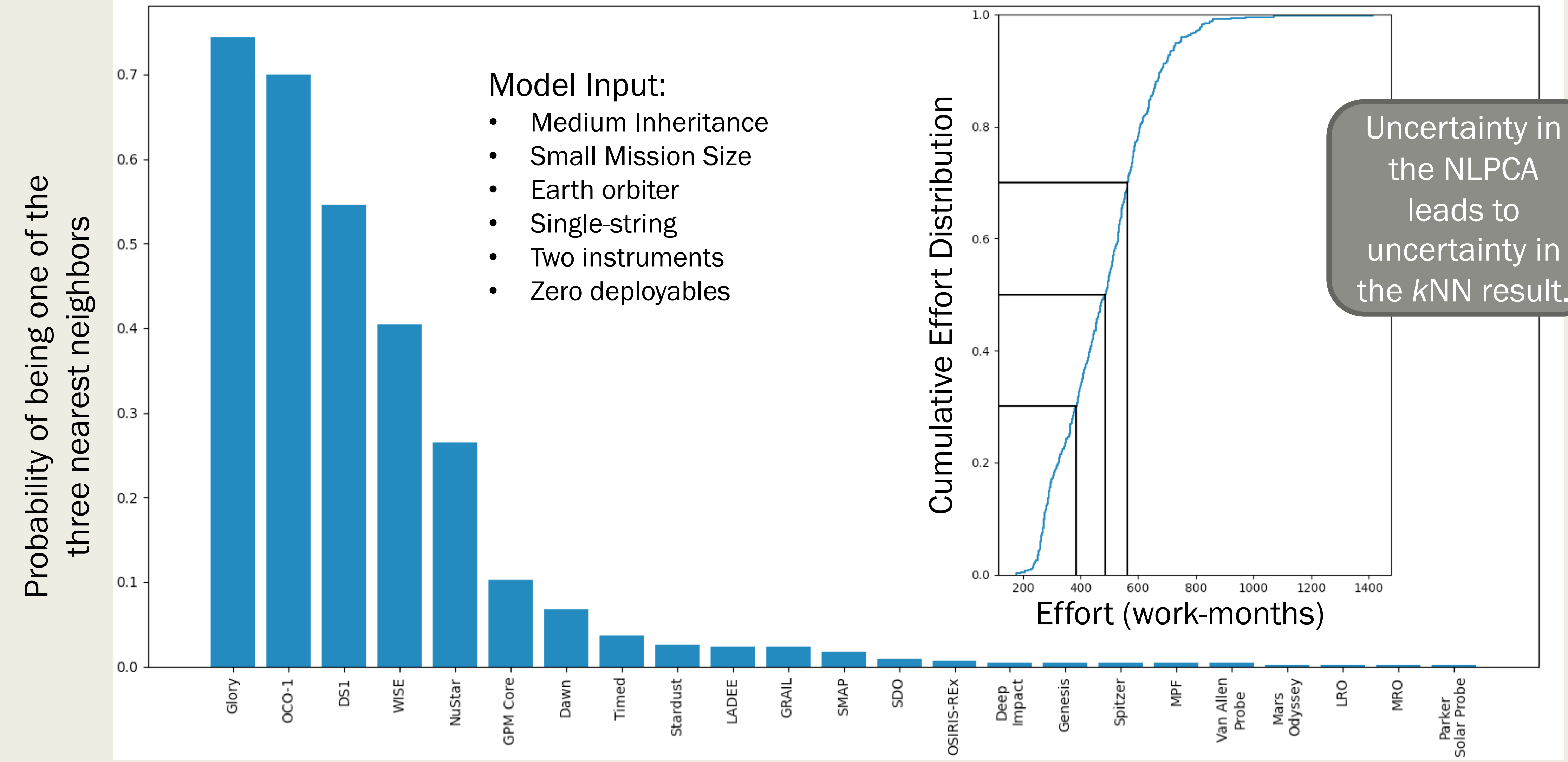
- Once we have our missions in a low-dimensional numeric space, we can calculate the distance from each mission to any model input easily (in a well-defined manner)
- If we choose k=2, we only use the closest two missions to generate an estimate.

Example
$k = 2$



$$\text{Cost(Your Project)} = \frac{\frac{\text{Cost}(\text{P}_1)}{d_1} + \frac{\text{Cost}(\text{P}_2)}{d_2}}{\frac{1}{d_1} + \frac{1}{d_2}}$$

*k*NN Model Example Output



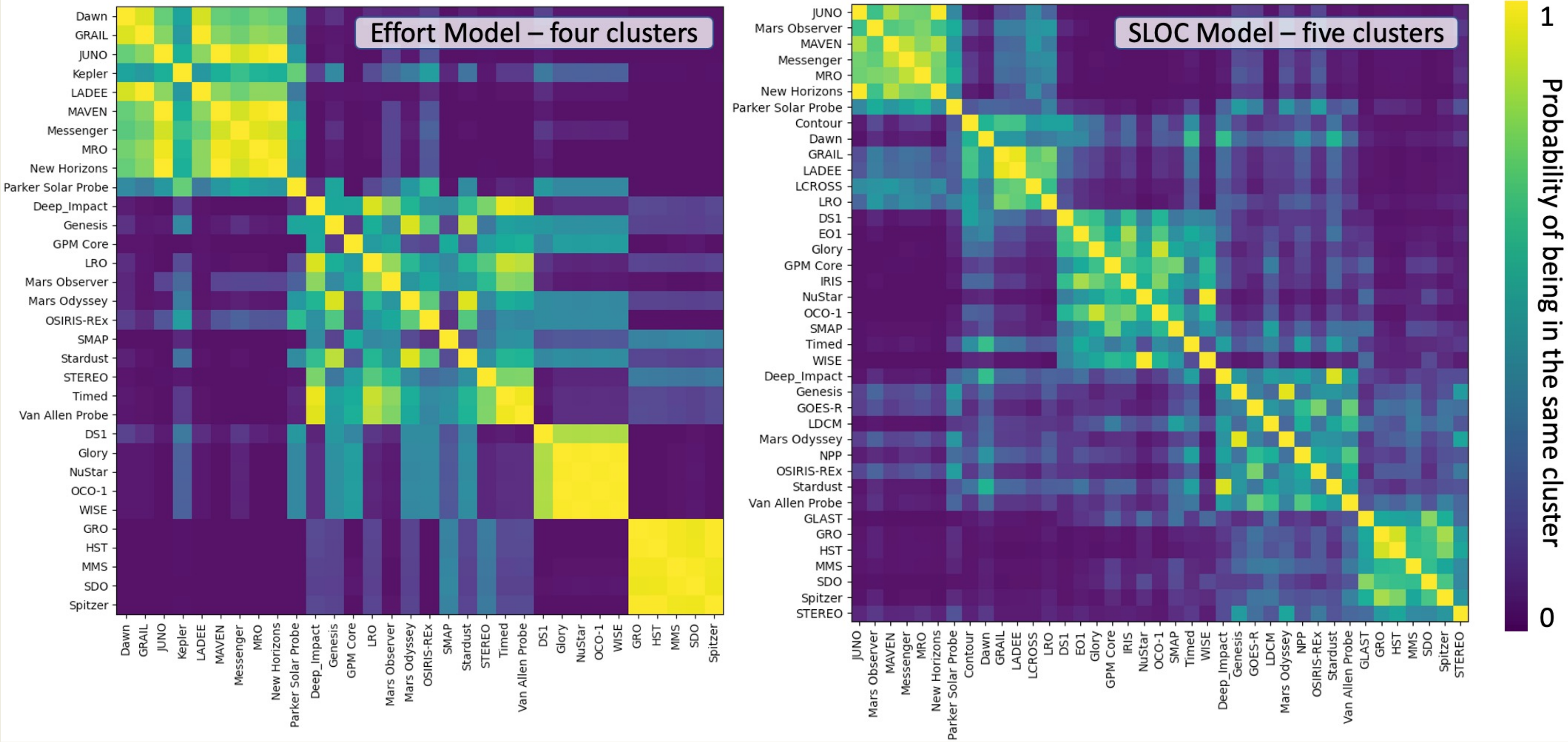
Uncertainty in the NLPCA leads to uncertainty in the *k*NN result.

NLPCA-based Clusters

Effort Model Clusters						
1. Very Large, Old, Outer Planetary	2. Rovers	3. Landers	4. Large, Complex, Inner-Outer Planetary	5. Large, Complex, Earth-Inner Planetary	6. Smaller, Higher Inheritance	7. Large, Earth Observatories and Constellations
Cassini	MER	Insight	Dawn	Deep Impact	DS1	GRO
Galileo	MPF	Phoenix	GRAIL	Genesis	GLORY	HST
	MSL		JUNO	GPM Core	NuStar	MMS
			Kepler	LRO	OCO-1	SDO
			LADEE	Mars Observer	WISE	Spitzer
			MAVEN	Mars Odyssey		
			Messenger	OSIRIS-REx		
			MRO	SMAP		
			New Horizons	Stardust		
			Parker Solar Probe	STEREO		
				TIMED		
				Van Allen Probe		

SLOC Model Clusters							
1. Very Large, Old, Outer Planetary	2. Rovers	3. Landers	4. Large, Complex, Inner-Outer Planetary	5. Large, Moderately Complex, Dual String (Cold)	6. Smaller or Simple, Earth – Asteroid/ Comet	7. Small-Medium, Single-String Inner-Planetary or Dual String (Cold) Asteroid/Comet	8. Large, Earth Observatories and Constellations
Cassini	MER	Insight	JUNO	Deep Impact	DS1	Contour	GLAST
Galileo	MPF	Phoenix	Mars Observer	Genesis	EO1	Dawn	GRO
	MSL		MAVEN	GOES-R	GLORY	GRAIL	HST
			Messenger	LDCM	GPM Core	LADEE	MMS
			MRO	Mars Odyssey	IRIS	LCROSS	SDO
			New Horizons	NPP	NuStar	LRO	Spitzer
			Parker Solar Probe	OSIRIS-REx	OCO-1		STEREO
				Stardust	SMAP		
				Van Allen Probe	TIMED		
					WISE		

Clustering Algorithm Overview



Probabilistic Linkage Matrices

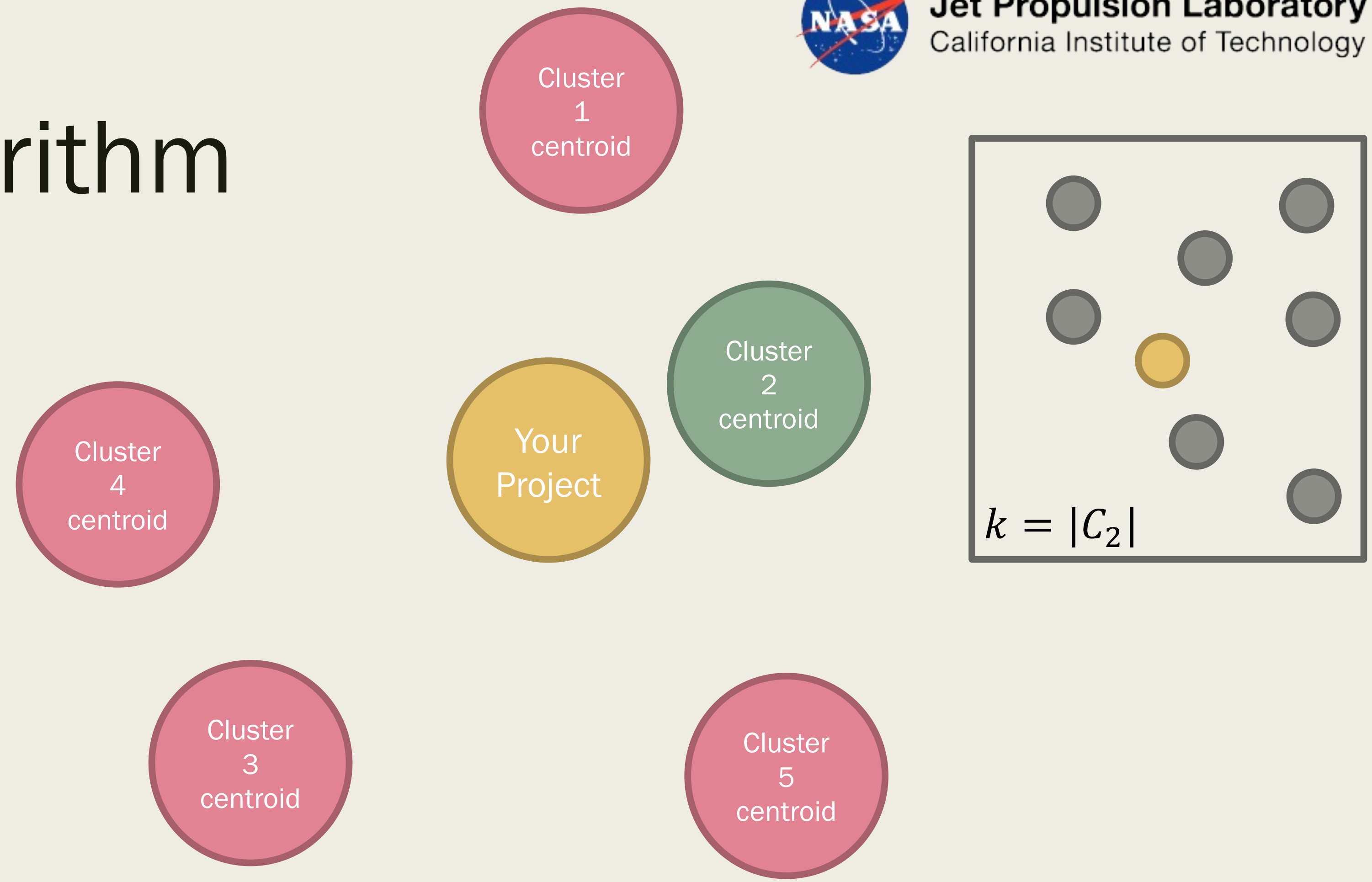
Calculated using the *k*-Means algorithm in NLPCA space

Cassini, Galileo, and Rovers and Landers are removed.

Jet Propulsion Laboratory
California Institute of Technology

Clustering Algorithm Overview

- Once we have our missions in a low-dimensional numeric space, we can calculate the distance from each mission to the "center" of any cluster
- Once in a cluster with *k* missions, use the *k*NN weighted average formula for the estimate.



$$\text{Cost(Your Project)} = \frac{\sum_{P \in C_2} \frac{Cost(P)}{d(P, \text{Your Project})}}{\sum_{P \in C_2} \frac{1}{d(P, \text{Your Project})}}$$

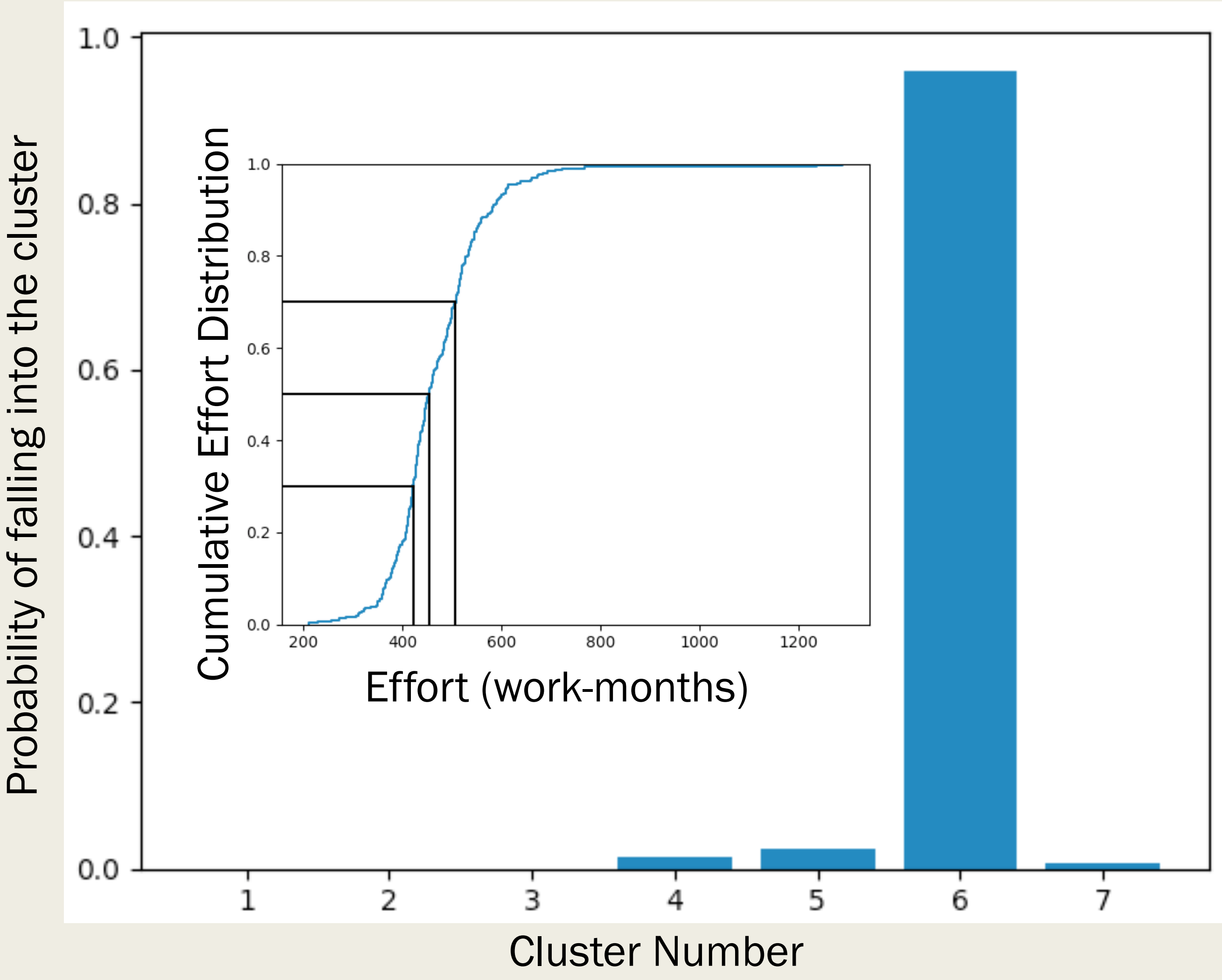
Jet Propulsion Laboratory
California Institute of Technology

Clustering Model Example Output

Model Input:

- Medium Inheritance
- Small Mission Size
- Earth orbiter
- Single-string
- Two instruments
- Zero deployables

Cluster 6 (Smaller, Higher Inheritance)
DS1
GLORY
NuStar
OCO-1
WISE



Uncertainty in the NLPCA leads to uncertainty in the cluster result.

Uncertainty in the Effort distribution is caused by uncertainty in the NLPCA as well as uncertainty in the cluster.

Stuff I Didn’t Really Talk About Today

- COCOMO (unchanged from previous ASCoT version)
- Model performance
  - *kNN and Clustering models perform at least as good as older version while also reporting uncertainty*
  - *CER performs significantly better than old version*
- ASCoT contains kNN and Clustering models for both Effort (in work-months) and SLOC (source lines of code)
- Website UI/UX is better than ever thanks to a major overhaul of the backend, allowing us to modularly and quickly build and deploy models using Dash
- The future of ASCoT includes models of Instrument Flight Software
- The future of ONSET includes your models too – let’s chat!

Closing Out

- COCOMO, CER, *k*NN, and Clustering models all produce probabilistic output
- CER tool reports uncertainty in model parameters
  - *Full posterior distribution available for download as a CSV*
- *k*NN and Clustering models utilize NLPCA and accounts for uncertainty in neural network fit
- All ASCoT and COMPACT models are available to use by navigating to ONSET within ONCE.
  - *oncedata.hq.nasa.gov*
- If you have a model you would like to publish online, shoot me an email!
  - *Samuel.R.Fleischer@jpl.nasa.gov.*