



# Fast, Unbiased Uncertainty Propagation with Multi-model Monte Carlo

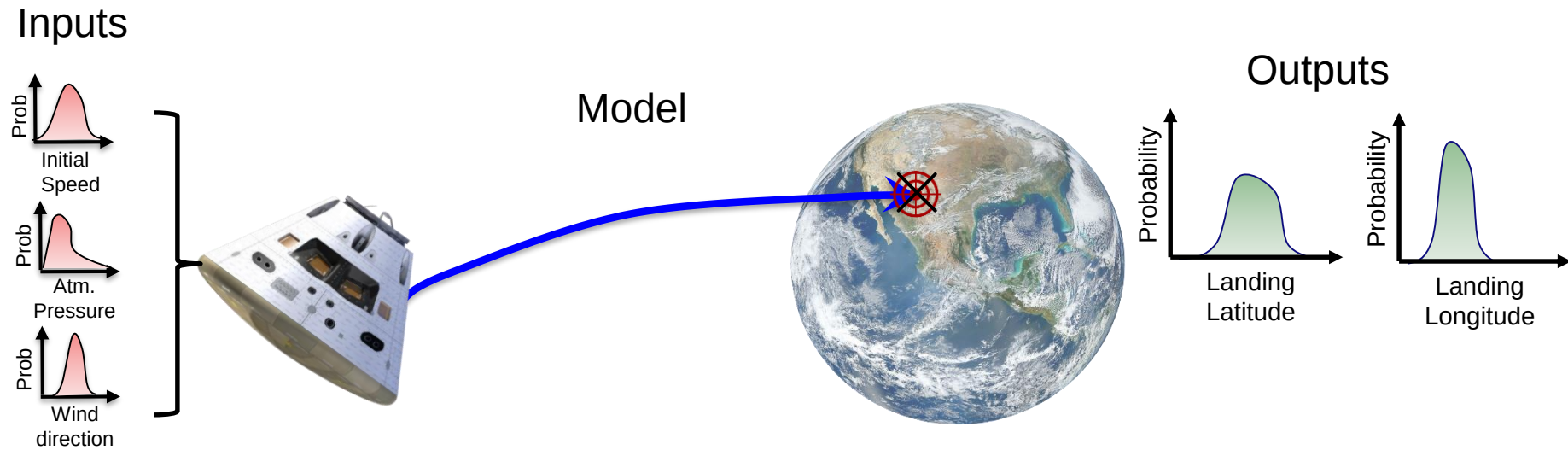
Geoffrey Bomarito<sup>1</sup>, James Warner<sup>1</sup>, Samantha Nieomoeller<sup>1,2</sup>,  
Luke Morrill<sup>1</sup>, Patrick Leser<sup>1</sup>, William Leser<sup>1</sup>, Robert A. Williams<sup>1</sup>,  
Soumyo Dutta<sup>1</sup>

1. NASA Langley Research Center, Hampton, VA

2. University of California – Los Angeles, Los Angeles, CA

# Uncertainty Propagation

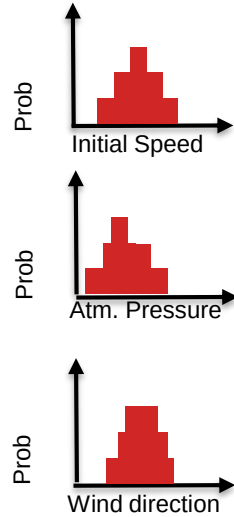
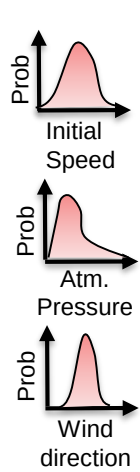
How does uncertainty in model inputs affect predictions of output?



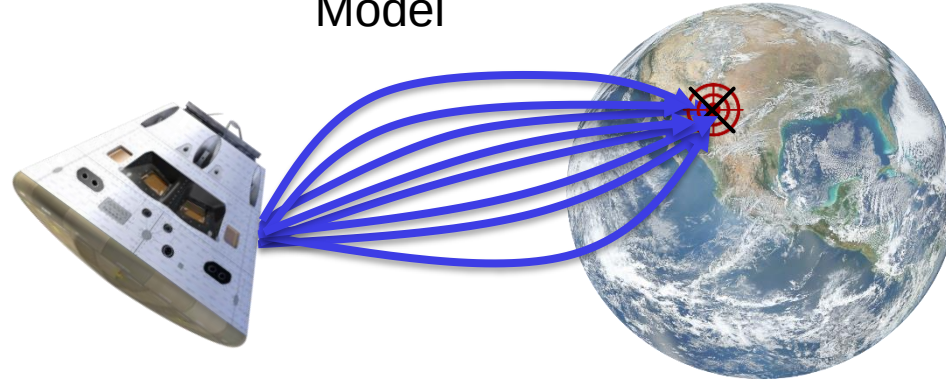
# Monte Carlo (MC) Simulation



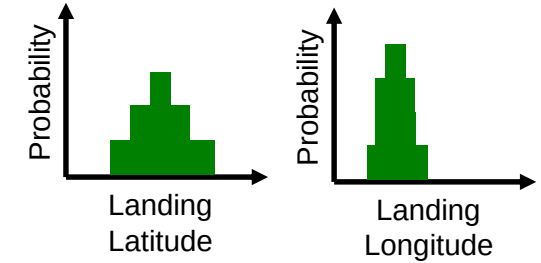
Inputs



Model



Outputs

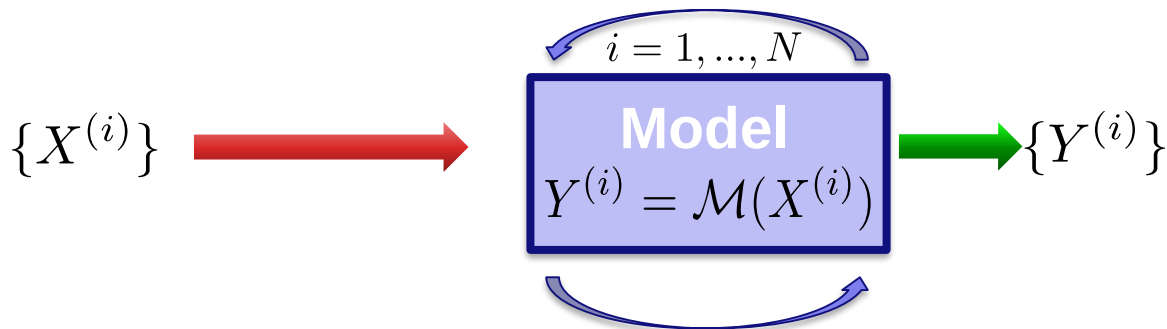


Let's estimate expected value (i.e. expected landing location) using MC

1) Draw  $N$  random inputs

2) Evaluate model

3) Form estimator



$$\mathbb{E}[Y] \approx \hat{Y} = \frac{1}{N} \sum_{i=1}^N Y^{(i)}$$

Unbiased:  $\mathbb{E}[\hat{Y}] = \mathbb{E}[Y]$

**Solution Time:  $N \times C$**

$N \sim O(10^4)$

HiFi model cost

Mean Squared Error = ~~bias~~ + variance

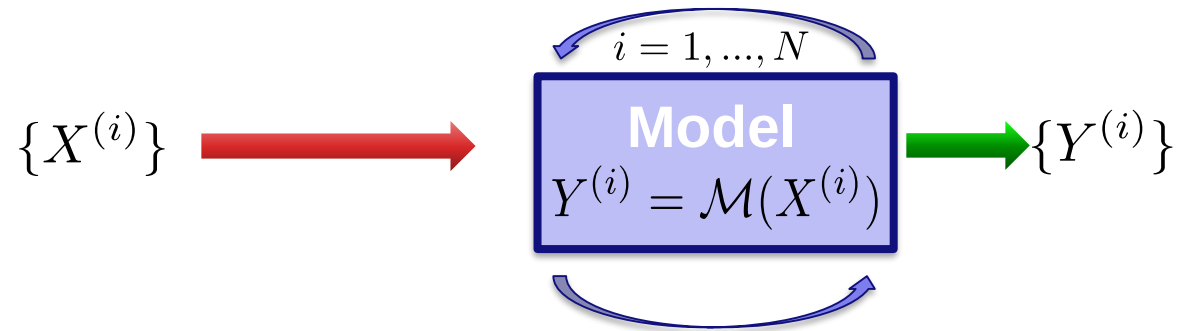
# Low-fidelity MC with Machine Learning (ML) Surrogates



1) Draw  $N$  random inputs

2) Evaluate model

3) Form estimator



$$\mathbb{E}[Y] \approx \hat{Y} = \frac{1}{N} \sum_{i=1}^N Y^{(i)}$$

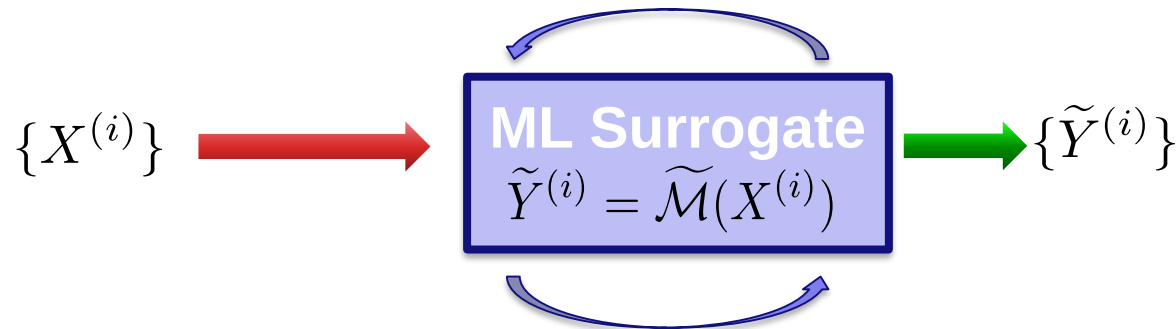
Unbiased:  $\mathbb{E}[\hat{Y}] = \mathbb{E}[Y]$

**Solution Time:**  $N \times C$

$N \sim O(10^4)$

Model cost

**How can we improve calculation time?**



$$\mathbb{E}[Y] \approx \hat{\tilde{Y}} = \frac{1}{N} \sum_{i=1}^N \tilde{Y}^{(i)}$$

Solution Time:  $N \times \tilde{C}$

Surrogate cost

Surrogate cost  $\ll$  Model cost

**Biased:**  $\mathbb{E}[\hat{\tilde{Y}}] \neq \mathbb{E}[Y]$

Mean Squared Error = bias + variance

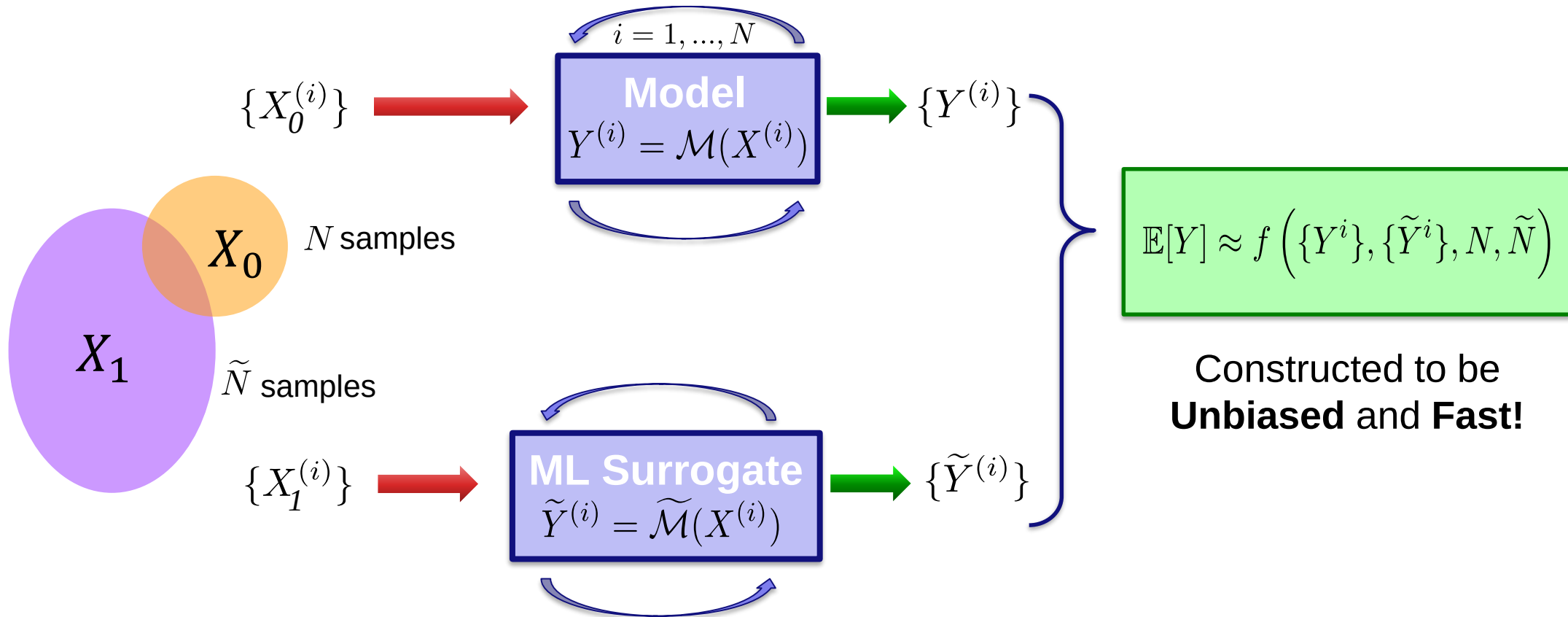
bias  $\rightarrow$  irreducible error!

# Multi-model Monte Carlo

1) Draw random inputs

2) Evaluate model

3) Form estimator



# Example: Multi-Level Monte Carlo MLMC<sup>[1]</sup>



$$\mathbb{E}[Y] \approx f\left(\{Y^i\}, \{\tilde{Y}^i\}, N, \tilde{N}\right)$$

**Main Idea:** Use a surrogate and correct with high-fidelity model

Unbiased

$$\mathbb{E}[\hat{Y}_{\text{MLMC}}] = E[Y]$$



Solution Time:

$$NC + (N + \tilde{N})\tilde{C}$$

$$\hat{Y}_{\text{MLMC}} = E[\tilde{Y}] + E[Y - \tilde{Y}]$$

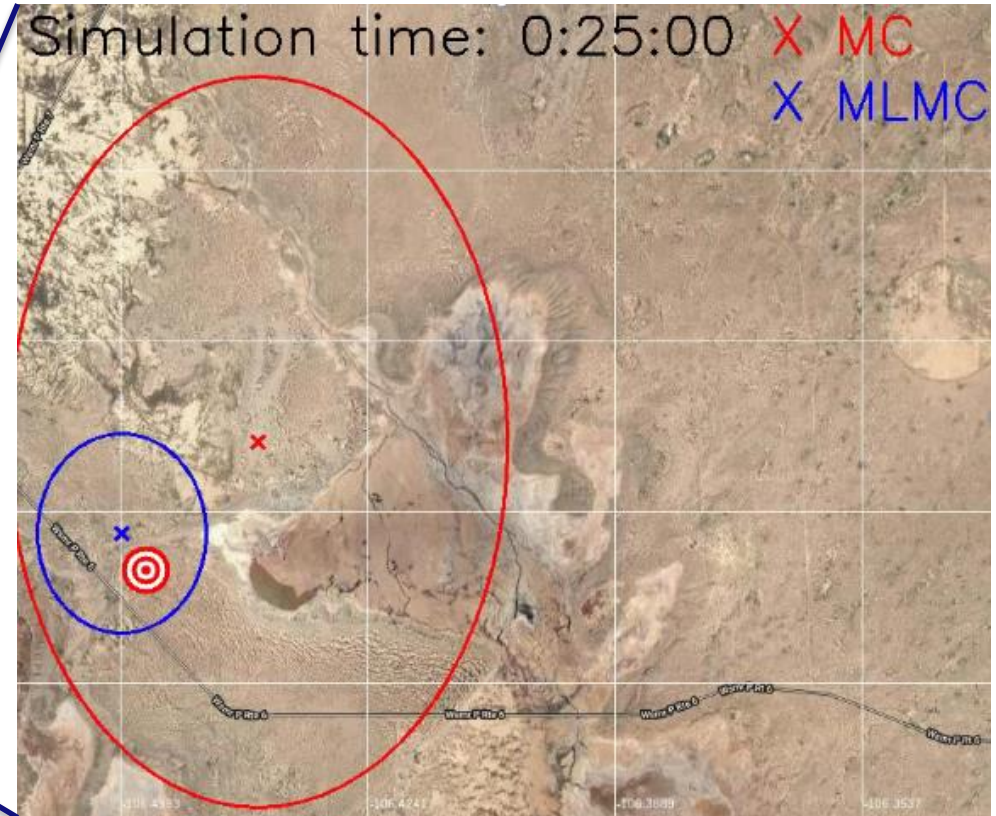
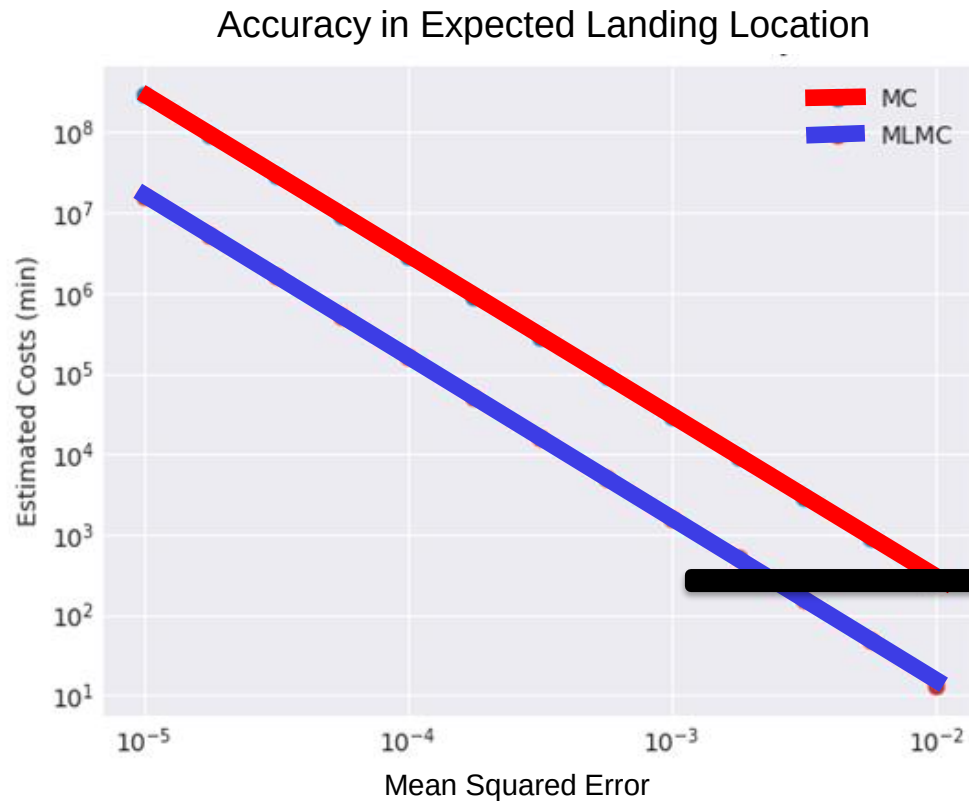
$$\frac{1}{\tilde{N}} \sum_{i=1}^{\tilde{N}} \tilde{Y}^i + \frac{1}{N} \sum_{i=1}^N (Y^i - \tilde{Y}^i)$$

Can Extend to More Models

Variance of difference of successive models should be decreasing

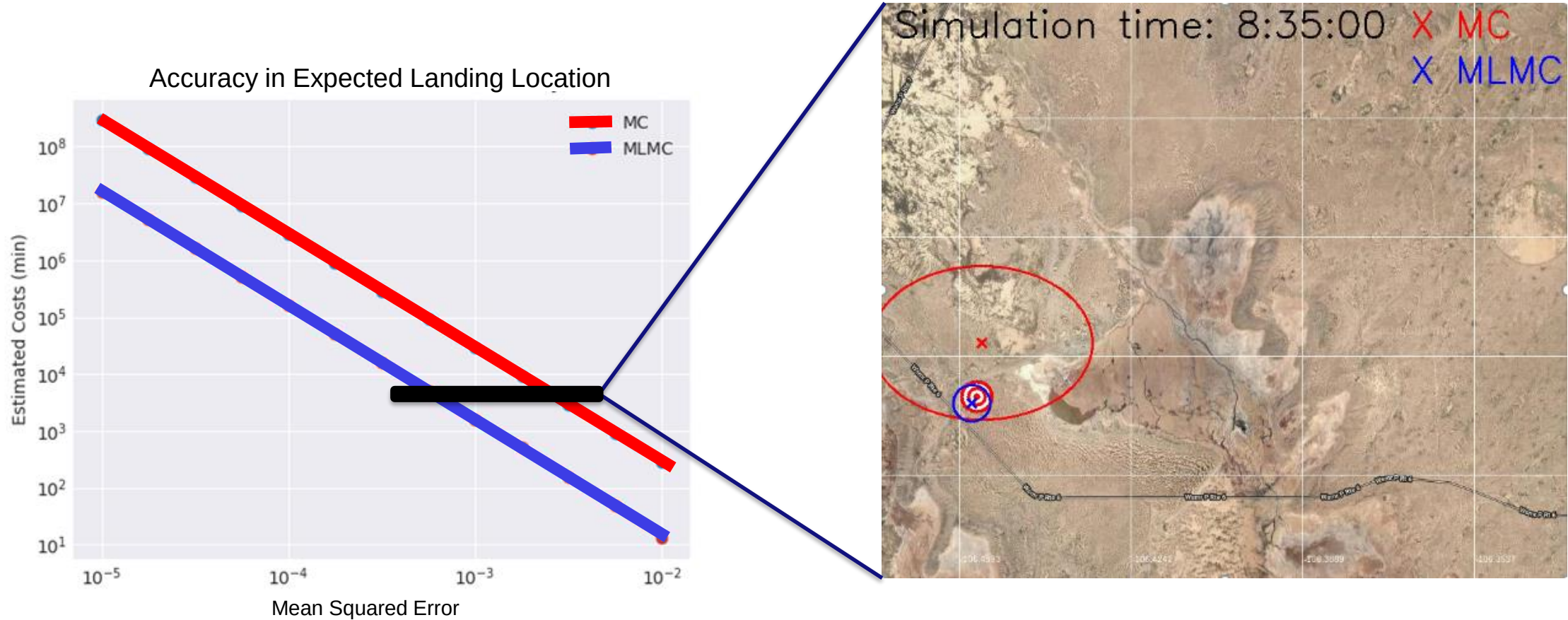
[1] Giles, M B. Operations Research (2008)

# MLMC Results



*More confidence with  
less time/resources*

# MLMC Results



*More confidence with  
less time/resources*

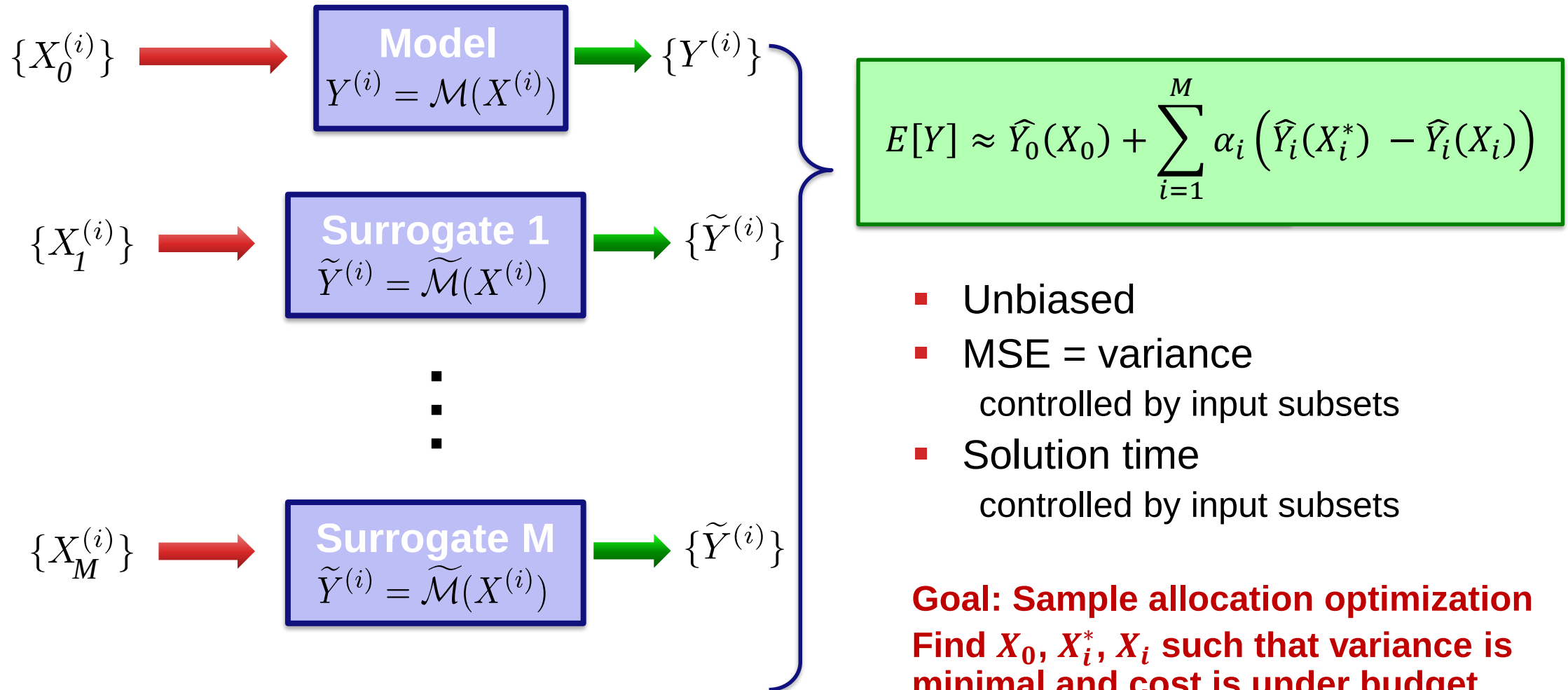
# Approximate Control Variates (ACV)<sup>[2]</sup>



1) Draw random inputs

2) Evaluate model

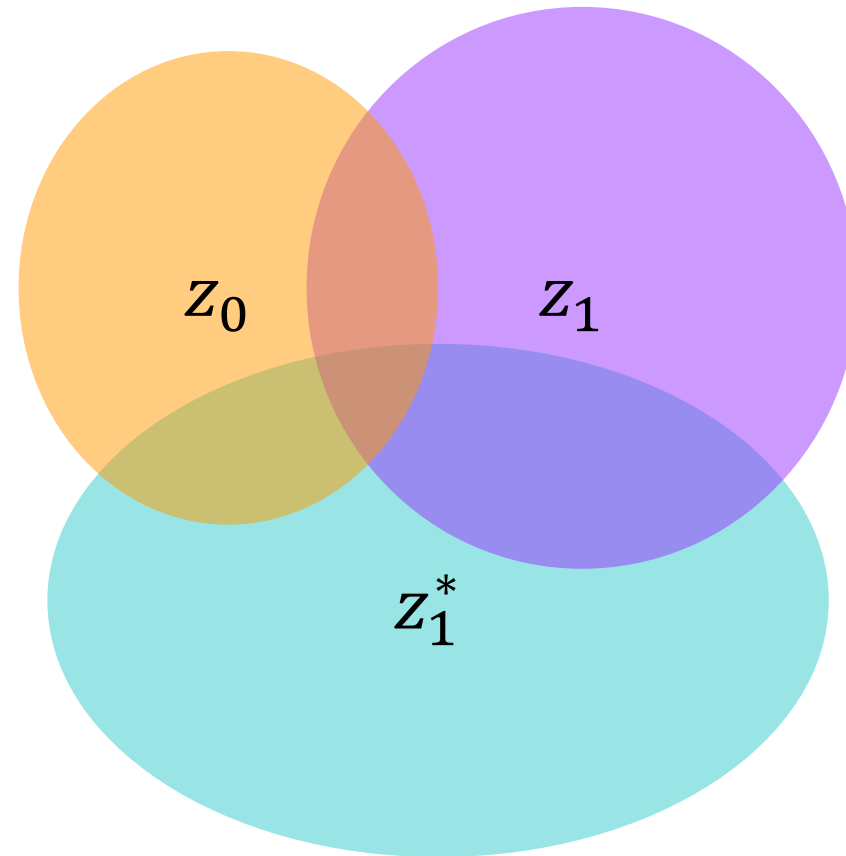
3) Form estimator



# Sample Allocation Optimization



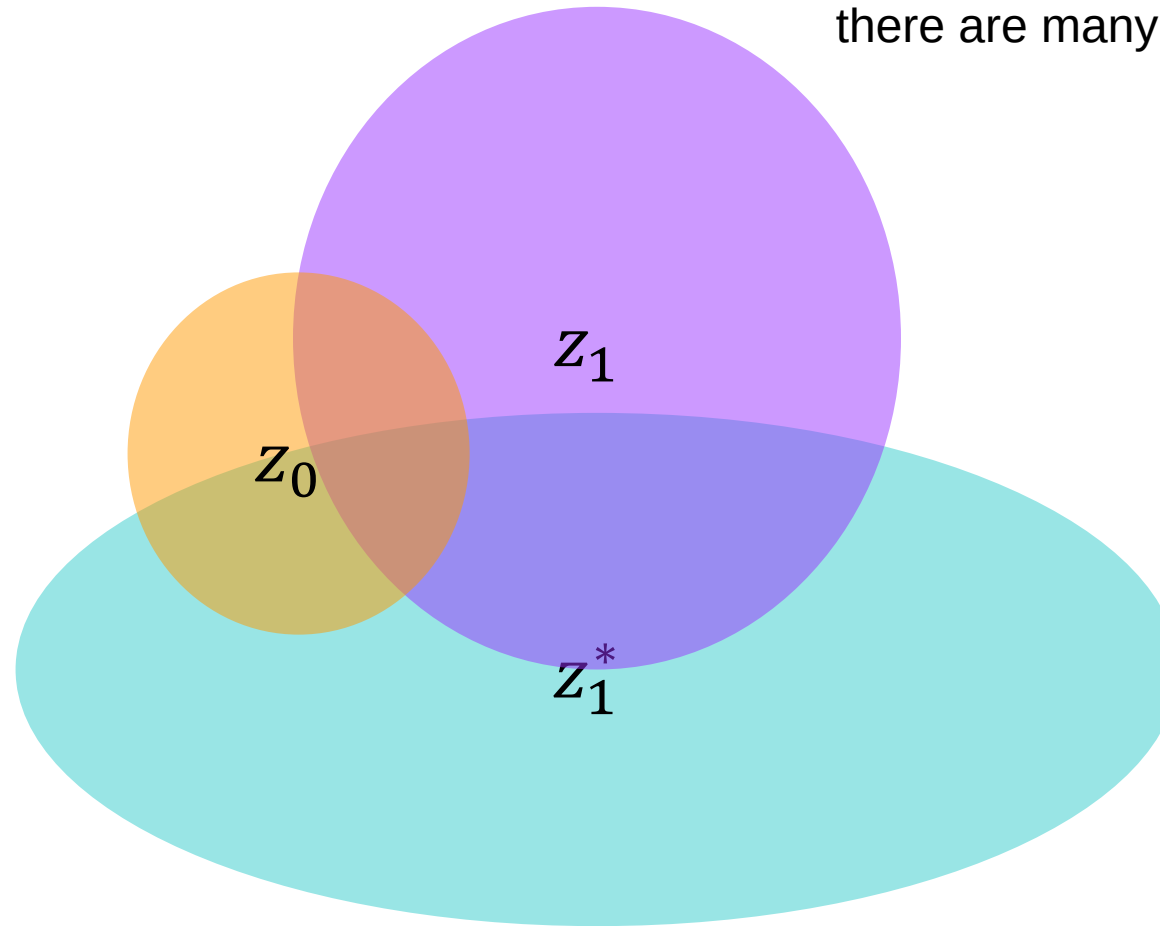
adjusting size and overlap changes MSE and cost



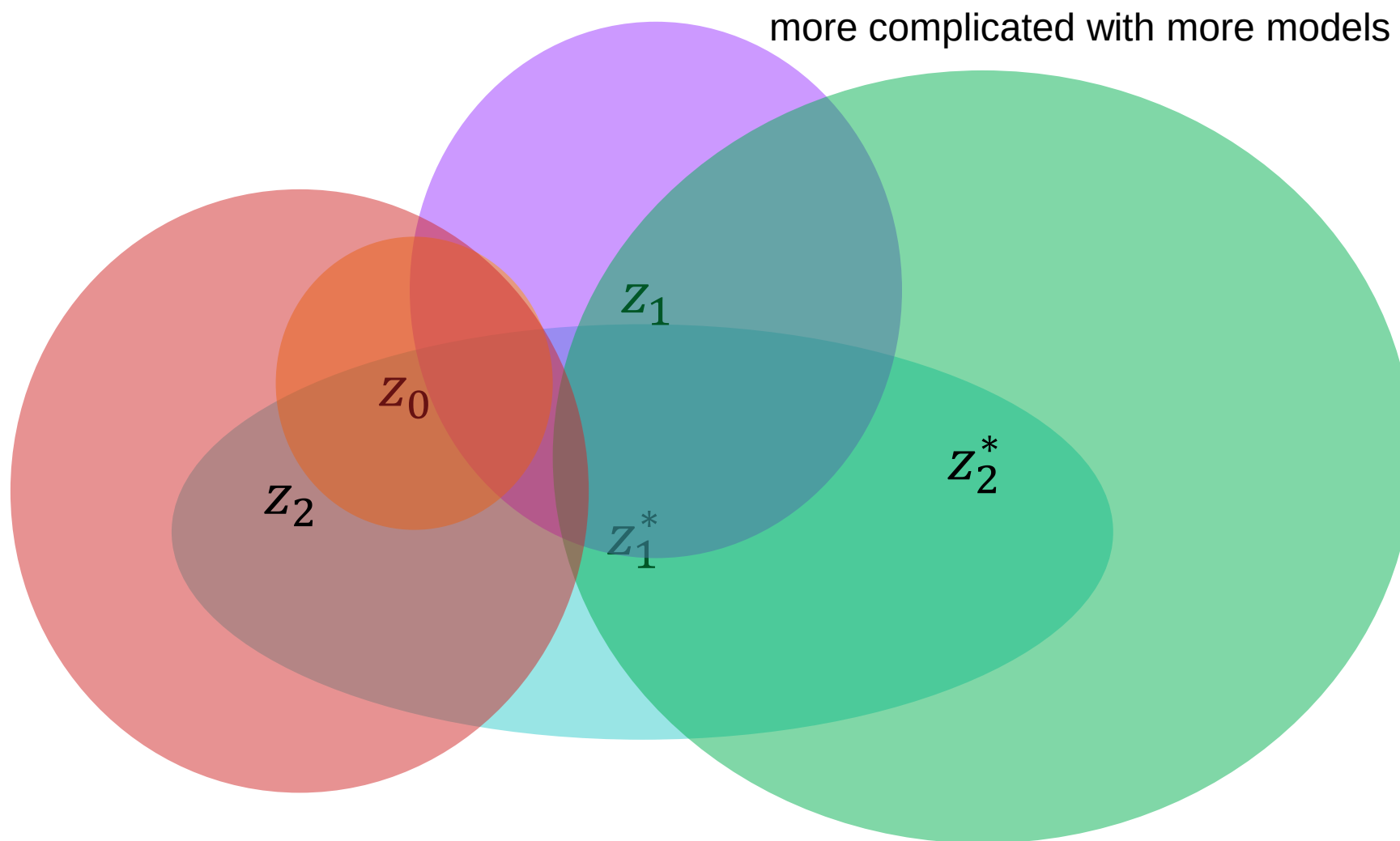
# Sample Allocation Optimization



there are many configurations



# Sample Allocation Optimization



**The domain of optimization is huge!**

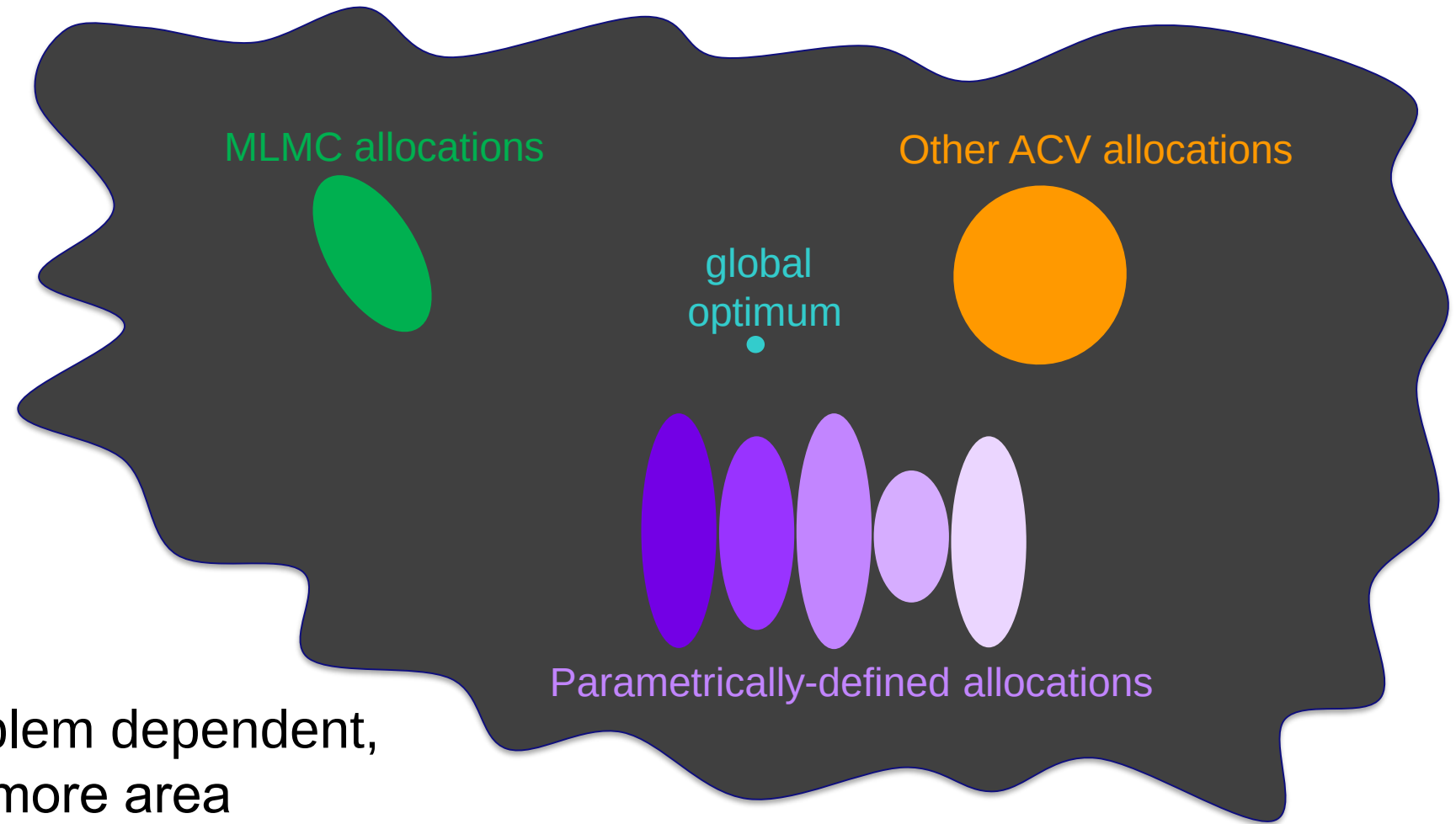
# Parametrically-defined ACV (PD-ACV) <sup>[3]</sup>



Domain of all possible  
ACV sample allocations

PD-ACVs are  
generalized versions of  
existing methods that  
cover more allocations

The best method is problem dependent,  
but it's better to search more area



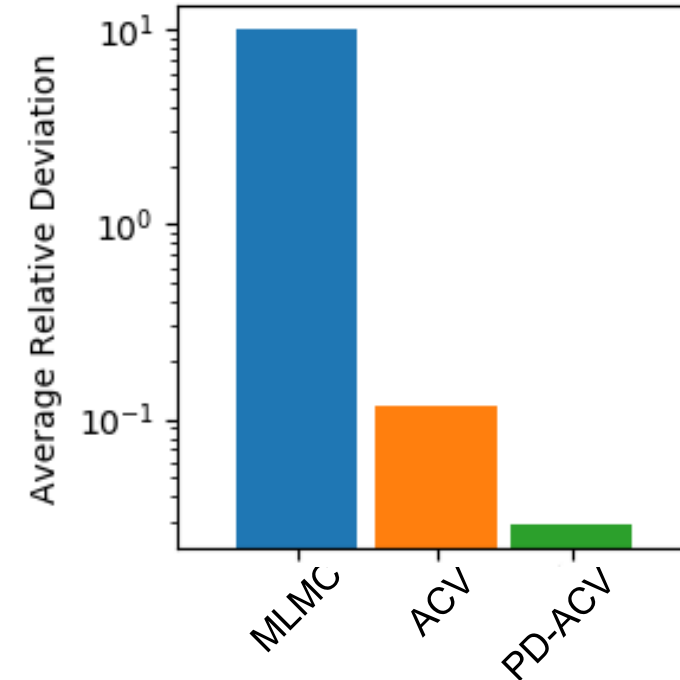
[3] Bomarito, Geoffrey F., et al. arXiv preprint arXiv:2012.02750 (2020).

# Random Model Scenarios



~5 Million

1. Choose number of models [2-6]
2. Sample random correlation matrix
3. Sample random variances
4. Sample random model costs (relative to hifi cost)
5. Perform allocation optimization for all algorithms (fixed cost budget)

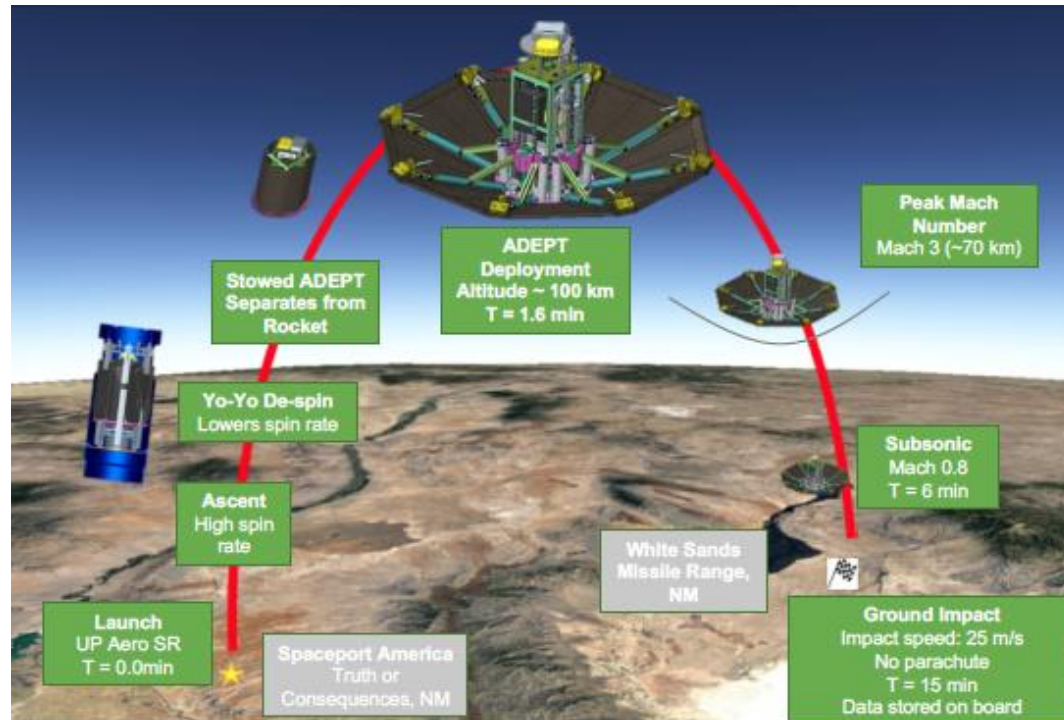


Results Moving Forward: Multi-Model MC = PD-ACV

# Application

Analysis of the Adaptable Deployable Entry & Placement Technology (ADEPT) Sounding Rocket 1 (SR-1) test flight

## ADEPT SR-1 Test Flight Schematic [4]



## Trajectory quantities of interest (QOIs) Considered

| Abbreviation | Description                                     |
|--------------|-------------------------------------------------|
| lat-td       | Geodetic latitude at touchdown                  |
| long-td      | Longitude at touchdown                          |
| rllrt-80km   | Roll rate at 80km geodetic altitude             |
| rllrt-60km   | Roll rate at 60km geodetic altitude             |
| alt-apo      | Apoapsis altitude                               |
| alt-mach1    | Altitude at Mach 1                              |
| alt-deploy   | ADEPT deploy altitude                           |
| time-td      | Touchdown time                                  |
| vel-term     | Terminal velocity                               |
| accel-max    | Max acceleration (Earth g's)                    |
| aoa-2km      | Total angle of attack at 2km geodetic altitude  |
| aoa-10km     | Total angle of attack at 10km geodetic altitude |
| aoa-35km     | Total angle of attack at 35km geodetic altitude |
| aoa-60km     | Total angle of attack at 60km geodetic altitude |
| aoa-80km     | Total angle of attack at 80km geodetic altitude |

- Program to Optimize Simulated Trajectories II (POST2) was used to model the flight trajectory
- The accuracy of multi-model MC Qol estimates was quantified versus both high- and low-fidelity MC

# Trajectory Simulation Models

## High-fidelity model:

- POST2 with time step  $\Delta t = 10^{-3}$  sec

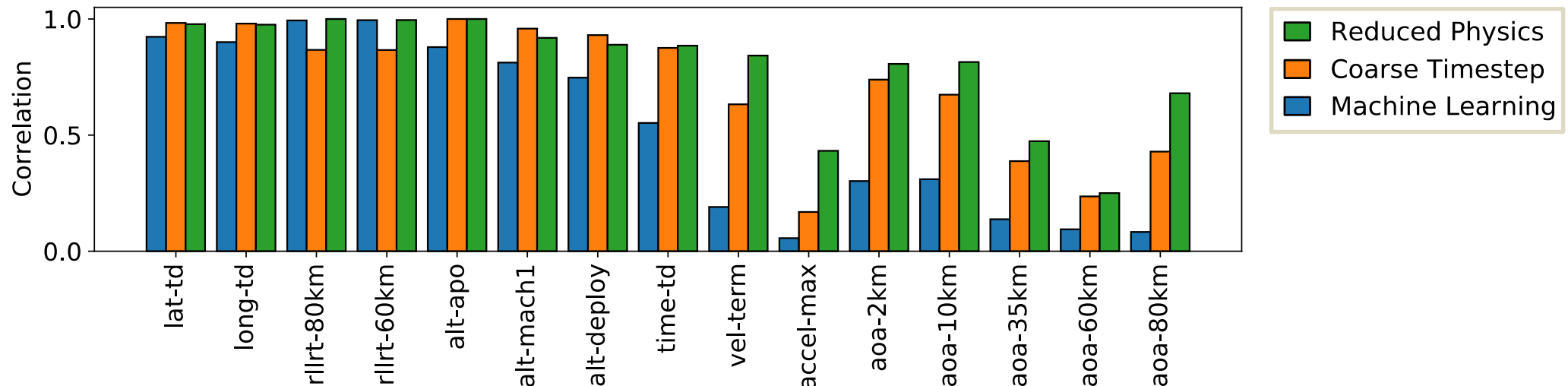
## Low-fidelity models:

1. Reduced physics: POST2 with simplified atmosphere model
2. Coarse time step: POST2 with  $\Delta t = 10^{-1}$  sec
3. Machine learning: Support vector machine (SVM) regression trained with input/output data from 250 POST2 simulations

## Trajectory Simulation Model Run Times

| Model            | Run Time (sec)       | Speedup (X)       |
|------------------|----------------------|-------------------|
| High Fidelity    | $2.2 \times 10^2$    | $1.0 \times 10^0$ |
| Reduced Physics  | $4.7 \times 10^1$    | $4.6 \times 10^0$ |
| Coarse Time Step | $2.8 \times 10^0$    | $7.8 \times 10^1$ |
| Machine Learning | $7.0 \times 10^{-4}$ | $3.1 \times 10^5$ |

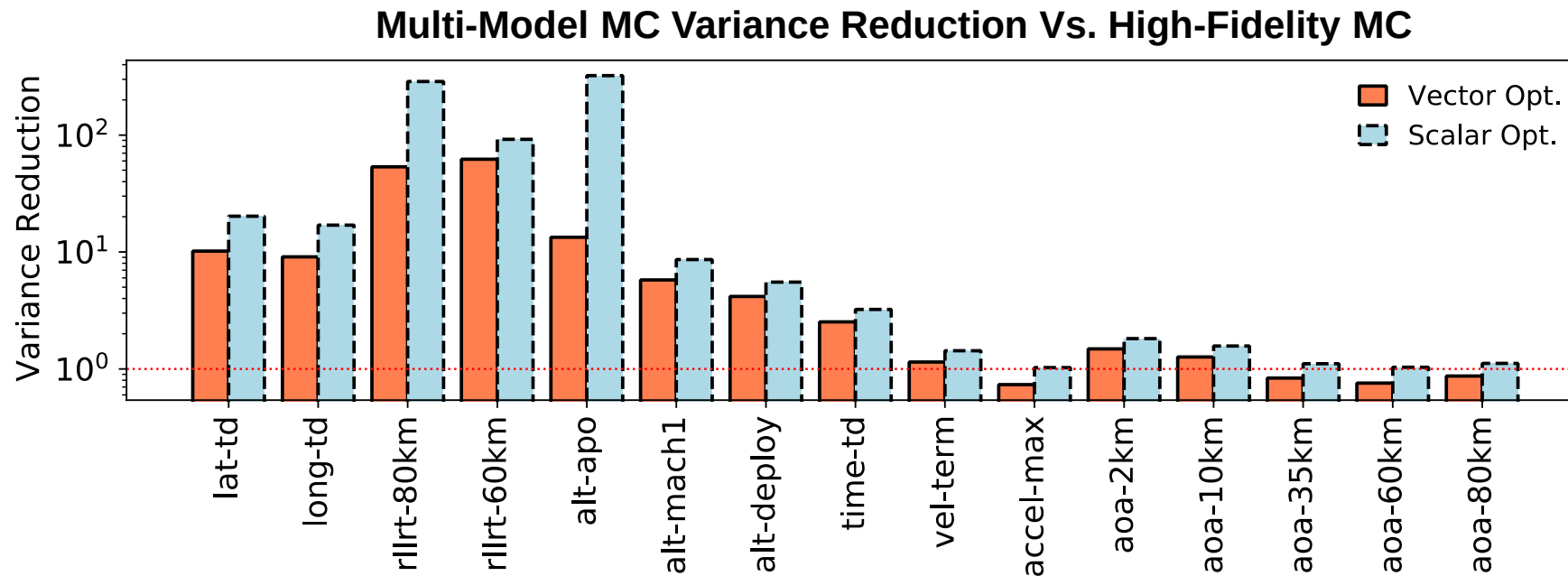
## Correlation of low-fidelity models with respect to the high-fidelity model



# Estimator Variance Comparison

For a computational budget of  $10^4$  seconds, the sample allocation optimization was solved for two cases

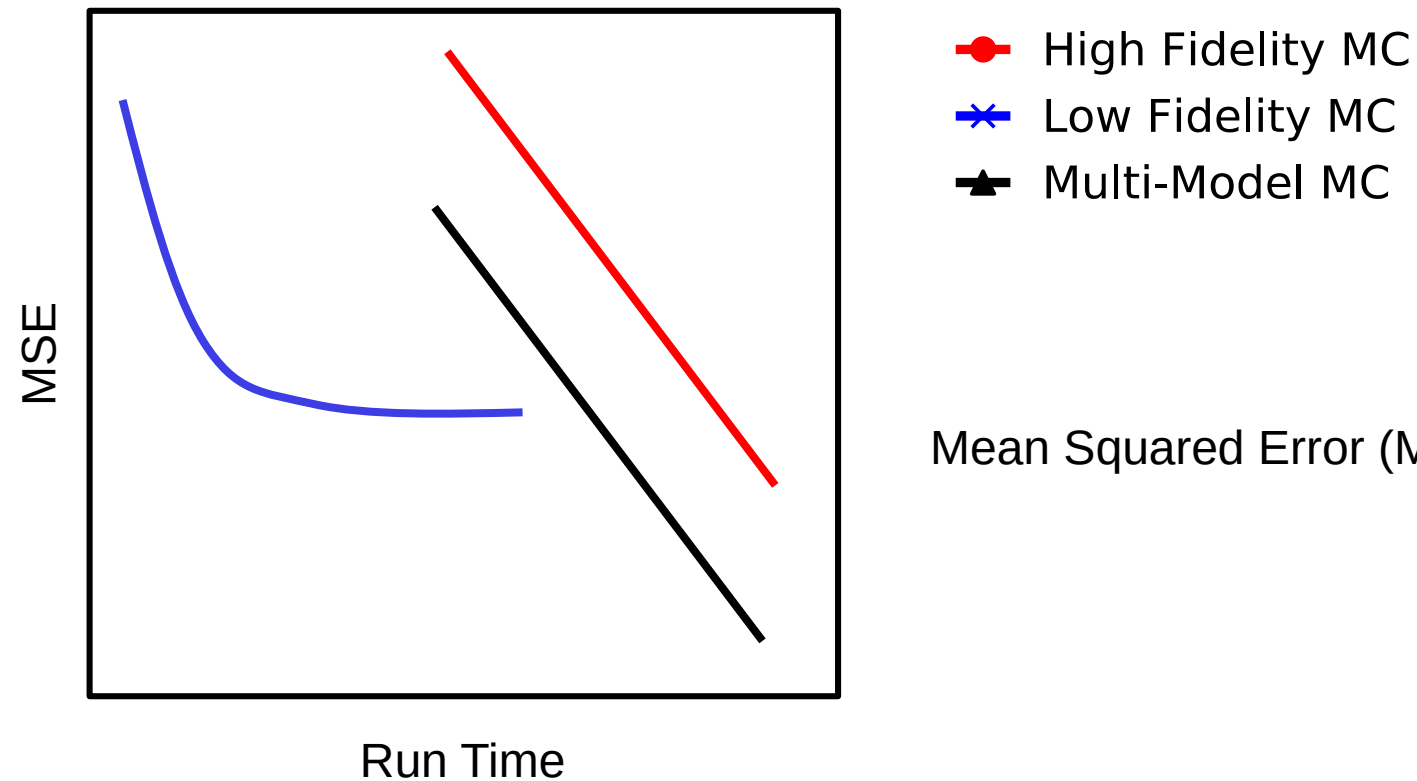
- **Vector Opt.:** minimizes variance for all QoIs simultaneously
- **Scalar Opt.:** minimizes variance for each QoI individually



# Estimator Accuracy vs. Run Time

- The MSE was calculated for high-, low-, and multi-model MC estimators versus a reference solution using high-fidelity MC with 100,000 samples

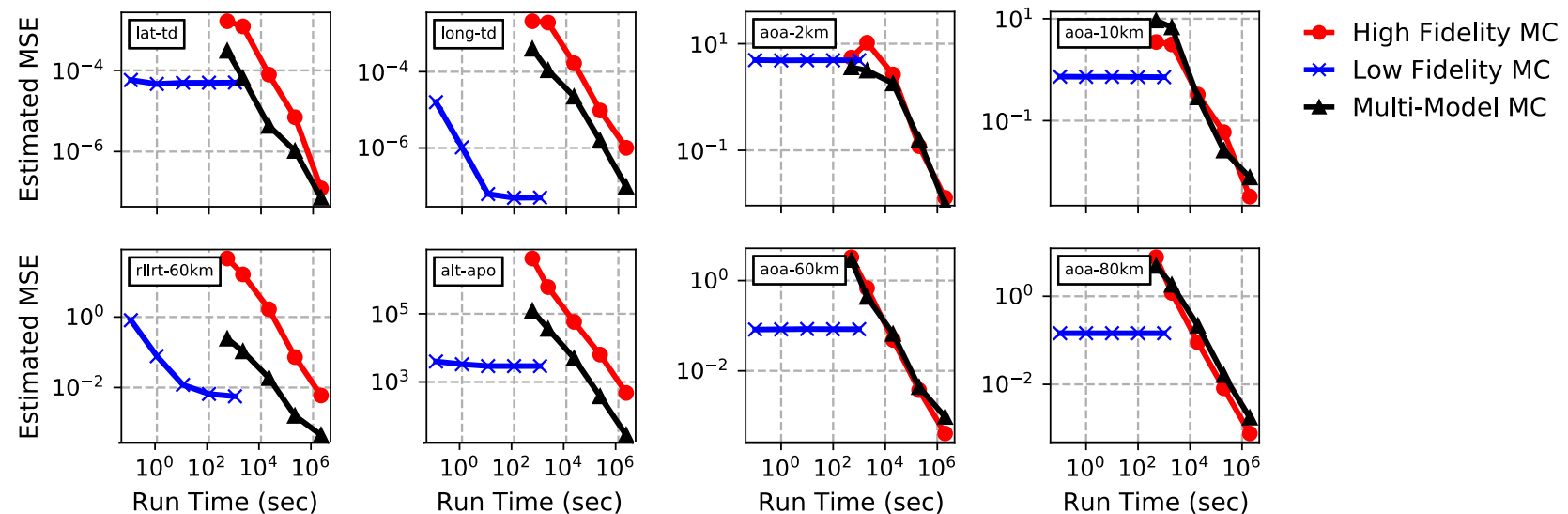
MSE versus Run Time for Several Trajectory QoIs



# Estimator Accuracy vs. Run Time

- The MSE was calculated for high-, low-, and multi-model MC estimators versus a reference solution using high-fidelity MC with 100,000 samples

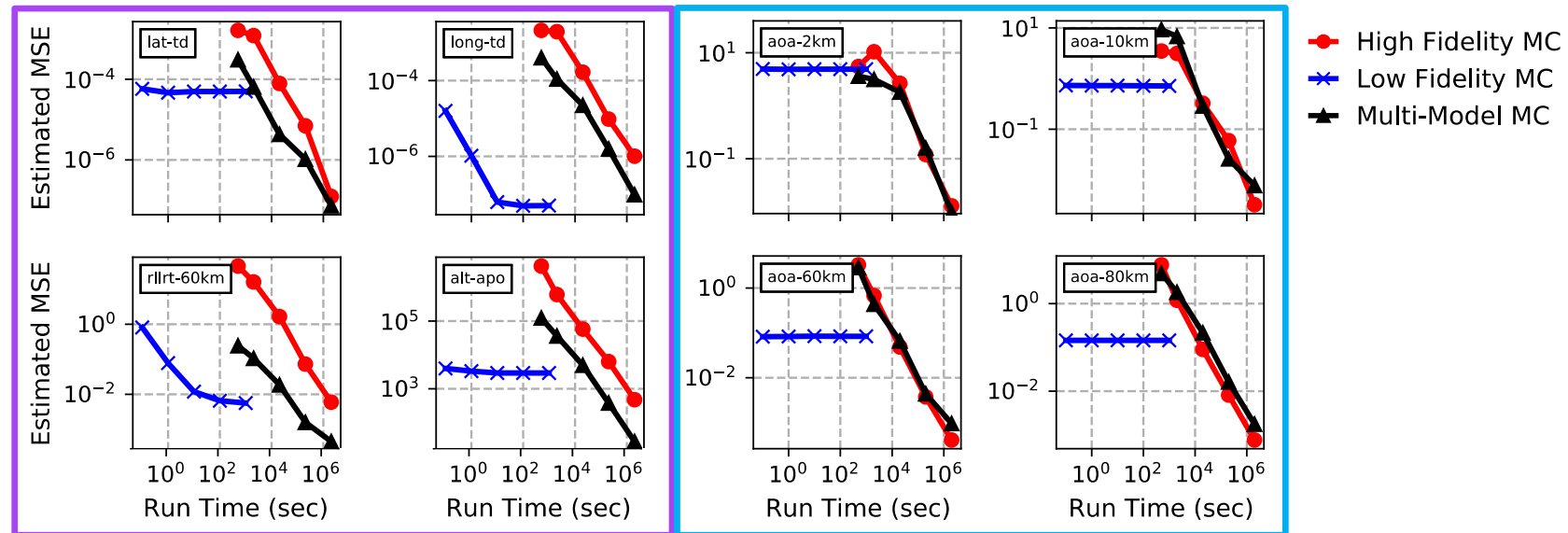
MSE versus Run Time for Several Trajectory QoIs



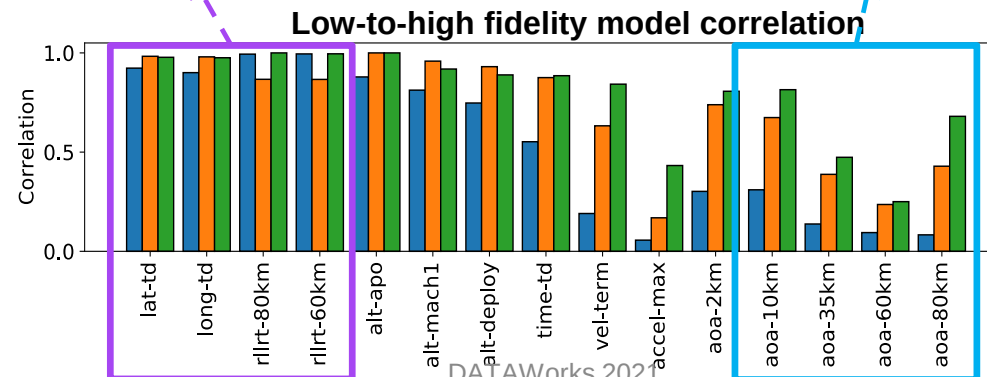
# Estimator Accuracy vs. Run Time

- The MSE was calculated for high-, low-, and multi-model MC estimators versus a reference solution using high-fidelity MC with 100,000 samples

MSE versus Run Time for Several Trajectory QoIs



Higher correlations yield large accuracy improvement relative to high-fidelity MC



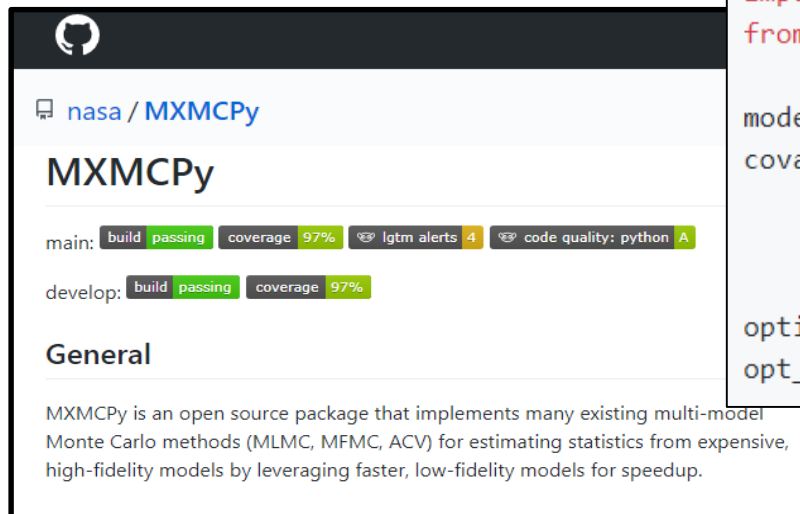
Lower correlations yield accuracy comparable to high-fidelity MC

# Conclusions & Future Work

- Multi-model MC approach leverages speed of surrogate models combined with accuracy (unbiasedness) of high-fidelity models
- Established the Multi-model MC approach to trajectory simulation as an alternative to standard MC that relies exclusively on a high- *or* low-fidelity model
- Demonstrated the potential to deliver up to two orders of magnitude accuracy improvement
- **Future Work:**
  - Develop low-fidelity models that have higher correlation with respect to the high-fidelity model for dynamic trajectory QoIs
  - Extend the multi-model MC approach to predict more advanced statistical quantities (probability distributions, rare event probabilities, landing ellipses, etc.)

# MXMCPy

- The best method is problem dependent
- [github.com/nasa/MXMCPy](https://github.com/nasa/MXMCPy)
- Simple interface to
  - compare and use different algorithms
  - implement new estimators



```
import numpy as np
from mxmc import Optimizer

model_costs = np.array([1.0, 0.05, 0.001])
covariance_matrix = np.array([[11.531, 11.523, 12.304],
                              [11.523, 11.518, 12.350],
                              [12.304, 12.350, 14.333]])

optimizer = Optimizer(model_costs, covariance_matrix)
opt_result = optimizer.optimize(algorithm="mfmc", target_cost=1000)
```



## Thank You

[geoffrey.f.bomarito@nasa.gov](mailto:geoffrey.f.bomarito@nasa.gov)

### References:

- [1] Giles, M. B., "Multilevel Monte Carlo path simulation," *Operations Research*, Vol. 56, No. 3, 2008
- [2] Gorodetsky, A., Geraci, G. , Eldred, M., and Jakeman, J. D., "A generalized approximate control variate framework for multifidelity uncertainty quantification," *Journal of Computational Physics*, Vol. 408, 2020.
- [3] Bomarito, G. F., Leser, P. E., Warner, J. E., and Leser, W. P., "On the optimization of approximate control variates with parametrically defined estimators," arXiv preprint arXiv:2012.02750 (2020).
- [4] Dutta, S., and Green, J. S., "Flight mechanics modeling and post-flight analysis of ADEPT SR-1," AIAA Aviation 2019 Forum, AIAA, Dallas, TX, 2019.