

CSU DESI Smoking Gun

Complete Reproduction Guide

Step-by-Step Instructions for Exact Replication

Version: 1.1.0

Date: February 3, 2026

CSU Research Team

1. Overview

This capsule contains everything needed to reproduce the CSU Beta Function Lock validation against DESI 2024 data. The analysis tests whether the CSU prediction $w_a = -4(1+w_0)$ is consistent with cosmological observations.

What this capsule contains:

- Canonical DESI constraint data (frozen inputs)
- Complete Python analysis code
- Publication-quality plot
- Full technical documentation
- Cryptographic integrity verification

What results you will reproduce:

- CSU $k=4$: 0.48σ consistent with DESI data (rigorous method)
- Λ CDM: 2.37σ tension with DESI data (rigorous method)
- Likelihood ratio: 49.6:1 favoring CSU over Λ CDM

System requirements:

- Python 3.8+ (recommended: 3.10+)
- pip or conda package manager
- 100 MB disk space minimum

2. Quick Start (5 Steps)

Step 1: Download and extract the ZIP

Step 2: Install dependencies: `pip install -r requirements.txt`

Step 3: Run the analysis: `python csu_rigorous_analysis.py`

Step 4: Verify outputs match provided results

Step 5: Check cryptographic integrity: `sha256sum -c SHA256SUMS.txt`

3. Detailed Step-by-Step Instructions

Step 1: Download the Capsule

Download `CSU_DESI_SmokingGun_Capsule_v1_FINAL_v2.zip` and save to your working directory.

Step 2: Extract the Archive

```
unzip CSU_DESI_SmokingGun_Capsule_v1_FINAL_v2.zip
cd CSU_DESI_SmokingGun_Capsule_v1_FINAL/
```


Step 3: Verify File Integrity (Recommended)

```
sha256sum -c SHA256SUMS.txt
```

Expected output: All files should show 'OK'

Step 4: Set Up Python Environment

```
# Option A: Using pip
pip install -r requirements.txt

# Option B: Using conda
conda create -n csu_desi python=3.10
conda activate csu_desi
pip install -r requirements.txt
```

Step 5: Run the Analysis

```
python csu_rigorous_analysis.py
```

Step 6: Verify Outputs

The script will generate:

- csu_rigorous_plot_reproduced.png (should match csu_rigorous_plot.png)
- csu_rigorous_results_computed.json (should match csu_rigorous_results.json)

```
# Compare outputs
diff csu_rigorous_results.json csu_rigorous_results_computed.json

# Hash comparison
sha256sum csu_rigorous_plot.png csu_rigorous_plot_reproduced.png
```


4. Understanding the Results

Two Methods Explained

This capsule provides two analysis methods. The **rigorous method** is the primary result; the **diagonal approximation** is provided for intuition.

Method 1: Rigorous (Full Covariance) — PRIMARY METHOD

- Uses complete covariance matrix Σ
- Accounts for strong correlation ($\rho=-0.897$) between w_0 and w_a
- Mahalanobis distance: $\chi^2 = (x-\mu)^T \Sigma^{-1} (x-\mu)$

Results:

- CSU $k=4$: $\chi^2_{\min} = 0.233$ (1 dof) $\rightarrow 0.48\sigma$ consistent
- Λ CDM: $\chi^2 = 8.04$ (2 dof) $\rightarrow 2.37\sigma$ tension
- Likelihood ratio: $\sim 49.6:1$ favoring CSU

Method 2: Diagonal Approximation — SUPPLEMENTARY

- Ignores correlation, uses only diagonal errors
- Simple z-score: $z = |\text{observed} - \text{predicted}| / \sigma$

Results:

- CSU $k=4$: $z \approx 0.16\sigma$ consistent
- Λ CDM: $z \approx 3.91\sigma$ tension

Note: This method is less rigorous but easier to understand.

Why Two Methods?

The strong negative correlation ($\rho=-0.897$) between w_0 and w_a creates a tilted confidence ellipse. The rigorous method accounts for this; the diagonal method does not. Both show CSU $k=4$ is consistent and Λ CDM is in tension, but with different magnitudes.

5. File Descriptions

File	Description	Purpose
desi_constraint.json	Canonical DESI data	Frozen inputs
csu_rigorous_analysis.py	Analysis code	Reproduces all results
requirements.txt	Python dependencies	Environment setup
CSU_Rigorous_Analysis.md	Full technical report	Complete documentation
SMOKING_GUN_SUMMARY.md	Executive summary	Quick overview
csu_rigorous_plot.png	Publication plot	Visualization
csu_rigorous_results.json	Numerical results	All computed values

MANIFEST.json	File inventory	Descriptions + hashes
SHA256SUMS.txt	Cryptographic hashes	Integrity verification
VERIFICATION_REPORT.md	Quality assurance	What was checked

6. Expected Numerical Results

Rigorous Method (Full Covariance):

```
 $\Lambda$ CDM Point (-1, 0):  
 $\chi^2$  = 8.04 (2 dof)  
p-value = 0.018  
 $\sigma$  equivalent = 2.37  
Status: Strong tension  
  
CSU k=4 Line ( $w_a = -4(1+w_0)$ ):  
 $\chi^2_{\min}$  = 0.233 (1 dof)  
p-value = 0.629  
 $\sigma$  equivalent = 0.48  
Status: Highly consistent  
  
Maximum Likelihood Ratio:  
CSU k=4 vs  $\Lambda$ CDM = 49.6:1
```

Diagonal Approximation:

```
CSU k=4:  
wa_predicted = -0.692  
wa_observed = -0.750  
delta = -0.058  
sigma_delta = 0.369  
z-score = 0.16 $\sigma$   
Status: Highly consistent  
  
 $\Lambda$ CDM:  
z_w0 = 2.76  
z_wa = 2.76  
z_total = 3.91 $\sigma$   
Status: Strong tension
```

7. Troubleshooting

Problem: SHA256 checksums don't match

Solution: Re-download the ZIP file, it may be corrupted

Problem: Python packages won't install

Solution: Try using a virtual environment or conda

Problem: Plot looks different

Solution: Check matplotlib version, try pip install matplotlib==3.7.0

Problem: Numbers slightly different (e.g., 0.481 vs 0.483)

Solution: This is normal due to floating-point precision, differences <1% are acceptable

8. Citation

If you use this analysis, please cite:

- DESI Collaboration, 'DESI 2024 VI: Cosmological Constraints from BAO', arXiv:2404.03002

- This capsule: CSU DESI Smoking Gun v1.1.0 (2026-02-03)

9. Mathematical Details

Mahalanobis Distance:

$$\chi^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

Line Distance Minimization:

For line $w_a = -k(1+w_0)$, parametrize as $\mathbf{m}(t) = [t, -k-kt]$

$$\text{Minimize: } \chi^2(t) = (\boldsymbol{\mu} - \mathbf{m}(t))^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - \mathbf{m}(t))$$

Diagonal Approximation:

$$z = |\text{observed} - \text{predicted}| / \sigma_{\text{total}}$$

where $\sigma_{\text{total}} = \sqrt{(\sigma_1^2 + \sigma_2^2)}$ for independent errors