

An approach to modeling linear and non-linear self-shading losses with pvlib

Will Hobbs¹, Kevin Anderson², Mark Mikofski³, Madison Ghiz³

¹Southern Company, ²Sandia, ³DNV

2024 PV Performance Modeling Workshop, May 6, 2024



Self-shade (row-to-row shade)

- About 40% of US utility-scale solar has self-shade losses
 - 25% thin-film tracking
 - 18% fixed tilt
 - 2/3 crystalline → 12% of solar has non-linear losses

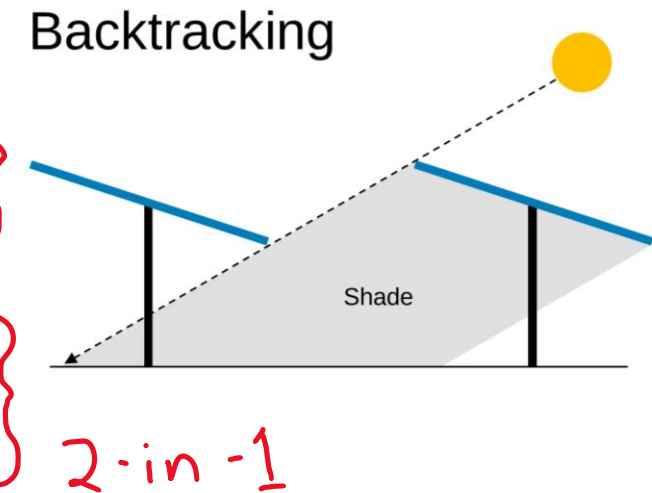
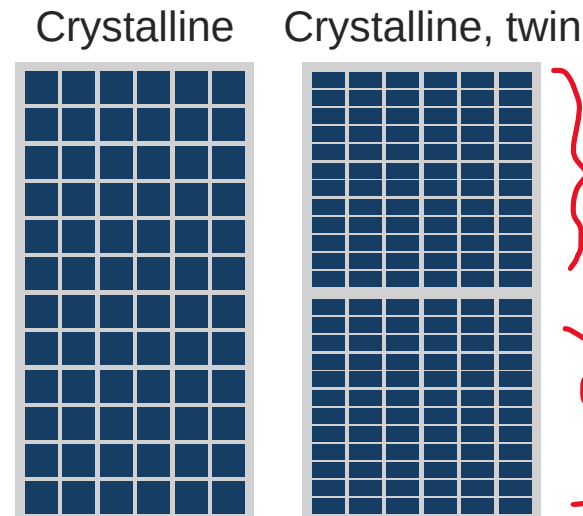
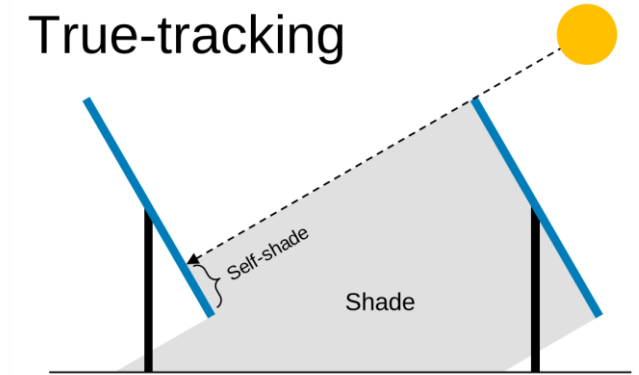
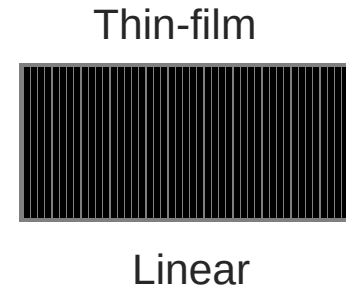
How do we model all of this?

Non-linearity

Twin modules (w/ half-cut cells)?

Fractional backtracking?

Tracker issues?



For example...

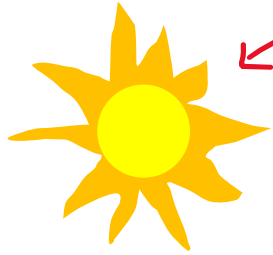
Easy^{ish} to model:

- Fixed crystalline w/ no shade



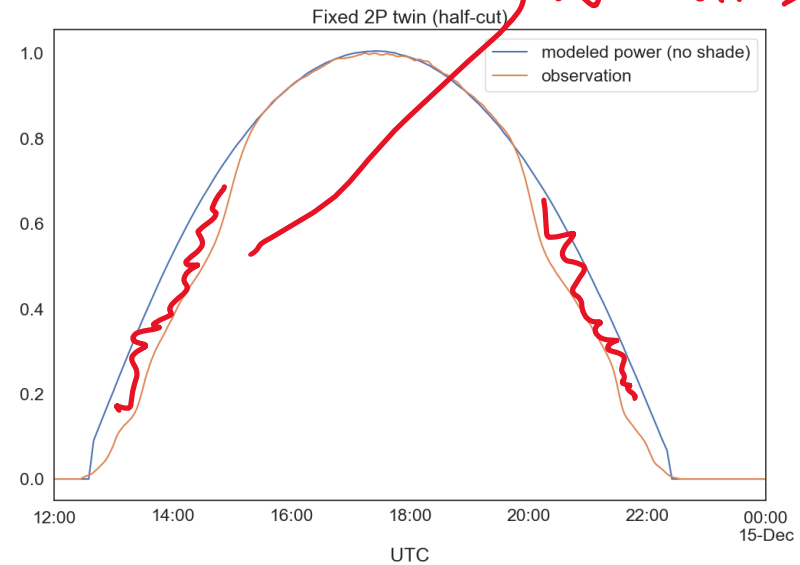
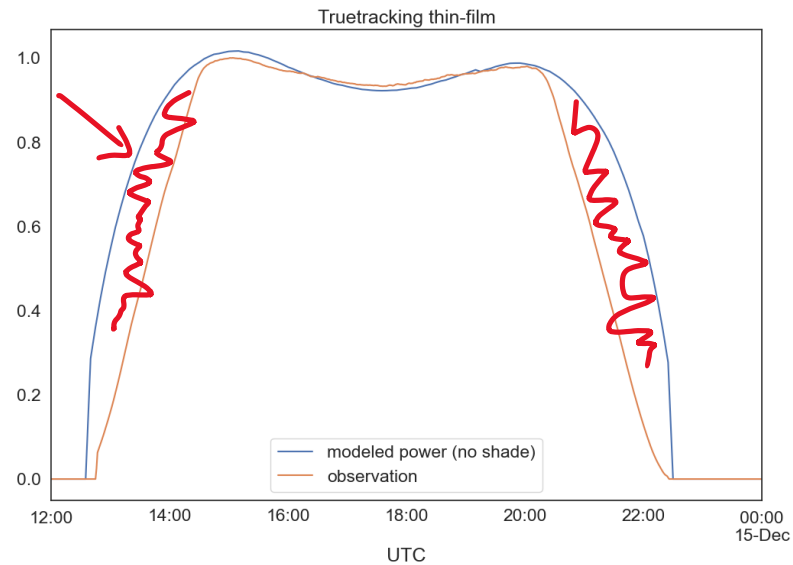
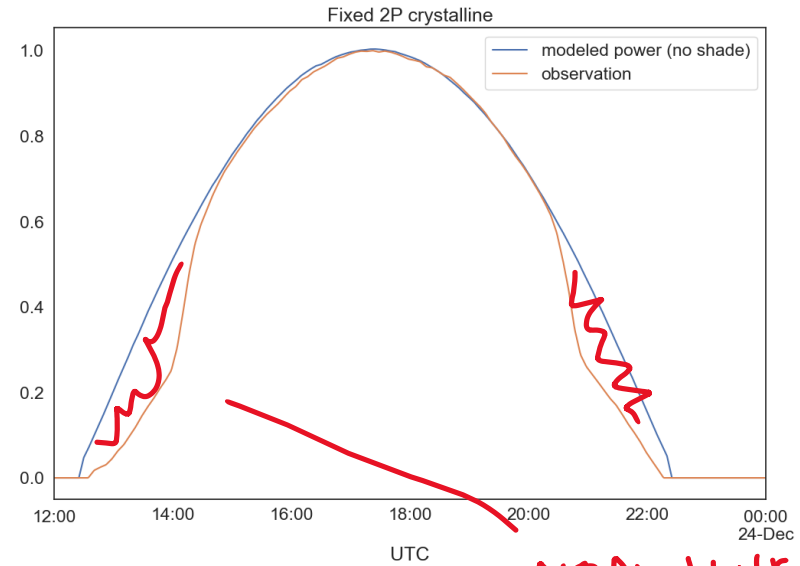
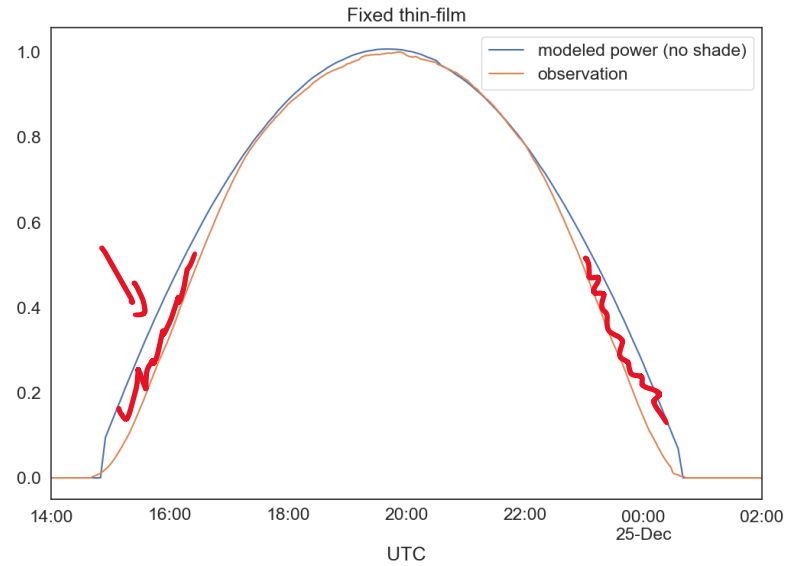
Hard^{-er} to model

- Lower sun elevation = shade
- Non-linear losses
- 2P, 2P twin, etc...



**What does this look
like for power?**

Overestimation at low sun angles:



NON-LINEAR

Solution using pvlb

Steps:

1. Calculate shaded fraction*
2. Calculate power loss**
3. Done!

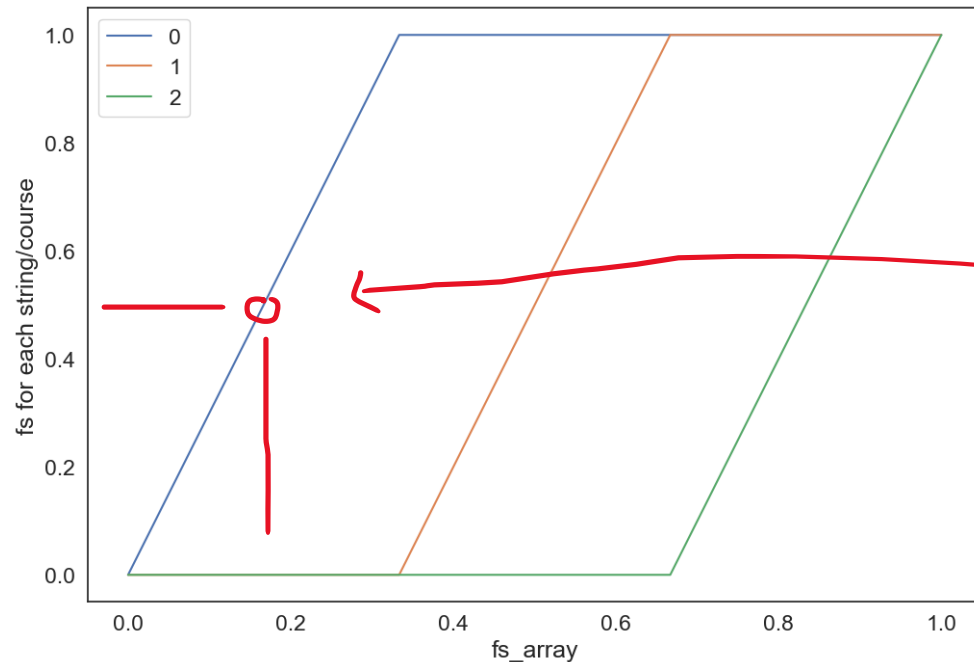
* Start with array shaded fraction, then calculate for each “course” (i.e., string) of modules

** actually, effective irradiance instead of power loss

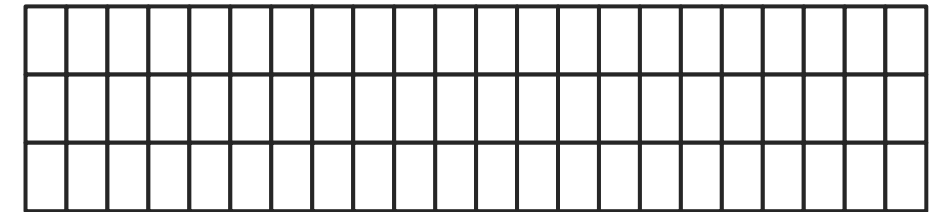
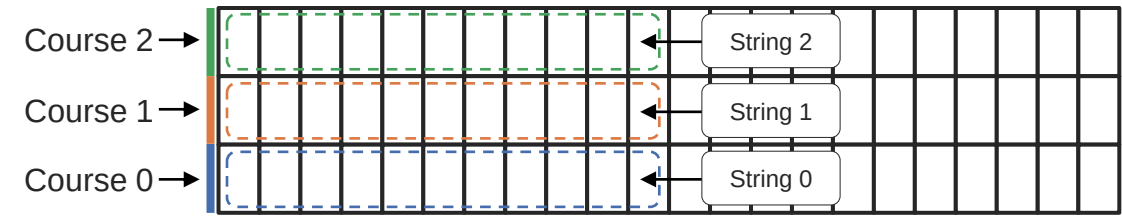
Shaded Fraction

- Use *shaded_fraction_front* from `infinite_sheds.get_irradiance()` for array shaded fraction
- Then calculate shaded fraction in each course^{*}

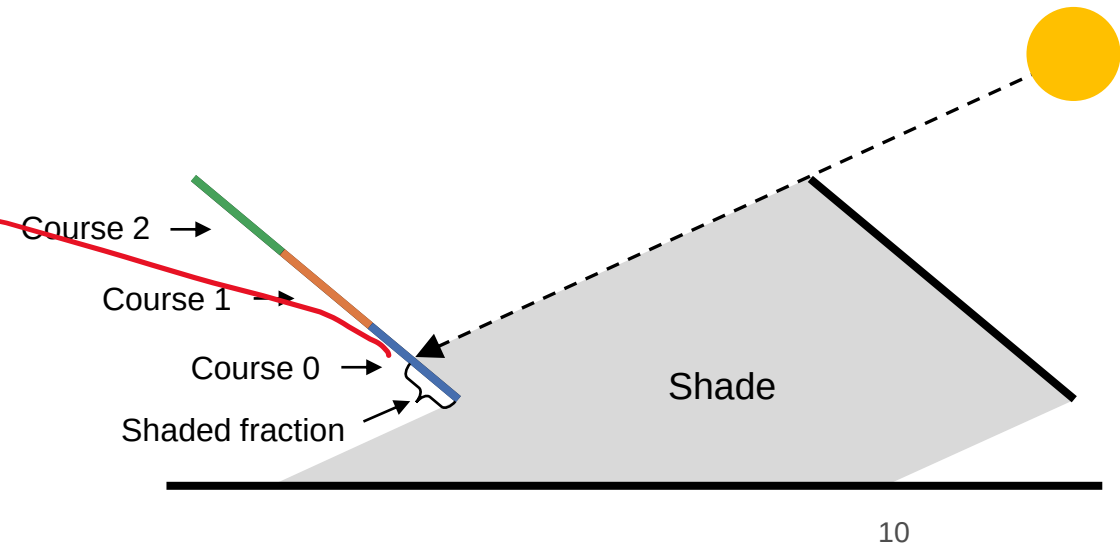
^{*} For our model, each string can only be in one course



Top-down view of 3P system (2 rows)



End view of 3P system (2 rows)



Twin modules?

- Double the number of effective modules up the side of a row
- For example, 2P twin \rightarrow 4P effective

Calculate power (effective irradiance) loss

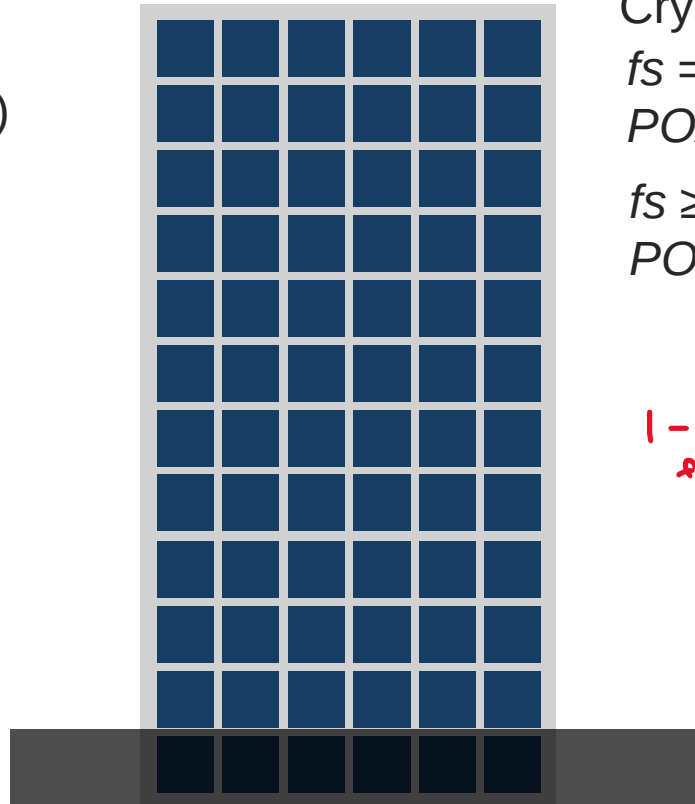
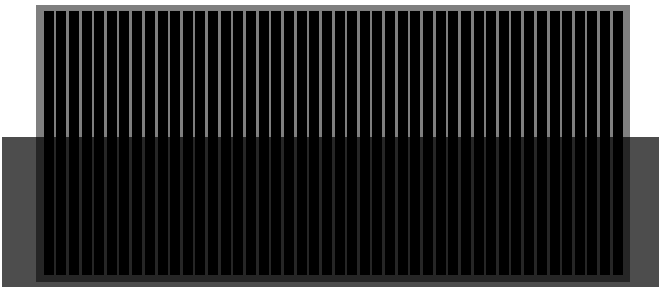
- Thin-film? Loss of direct irradiance linear with shaded fraction, f_s
- Crystalline? Loss of direct irradiance *across the first row of cells*

(and treat twin modules as 2 modules)

Thin-film (linear):

$$f_s = 1/2$$

POA_{direct} reduced by 1/2



Crystalline (non-linear):

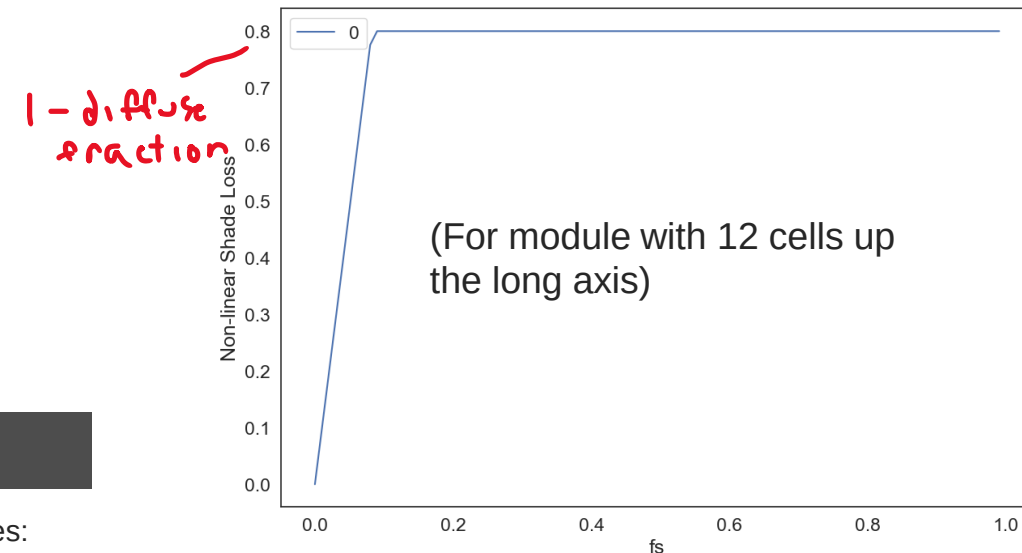
$$f_s = 1/24$$

POA_{direct} reduced by 1/2

$$f_s \geq 1/12$$

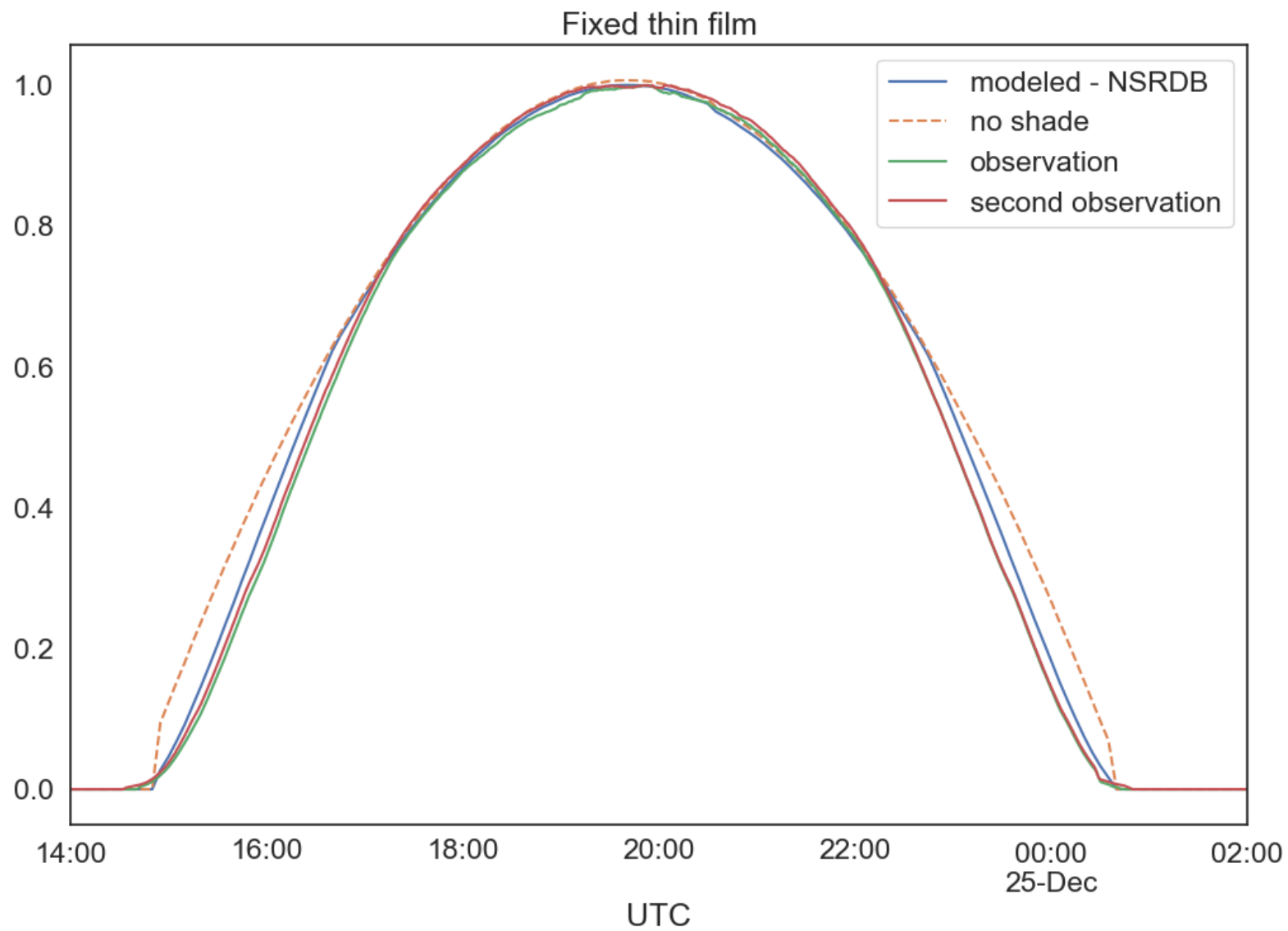
POA_{direct} reduced by 100%

1/2 of a cell
module is limited to cells w/ most shade (in portrait...)

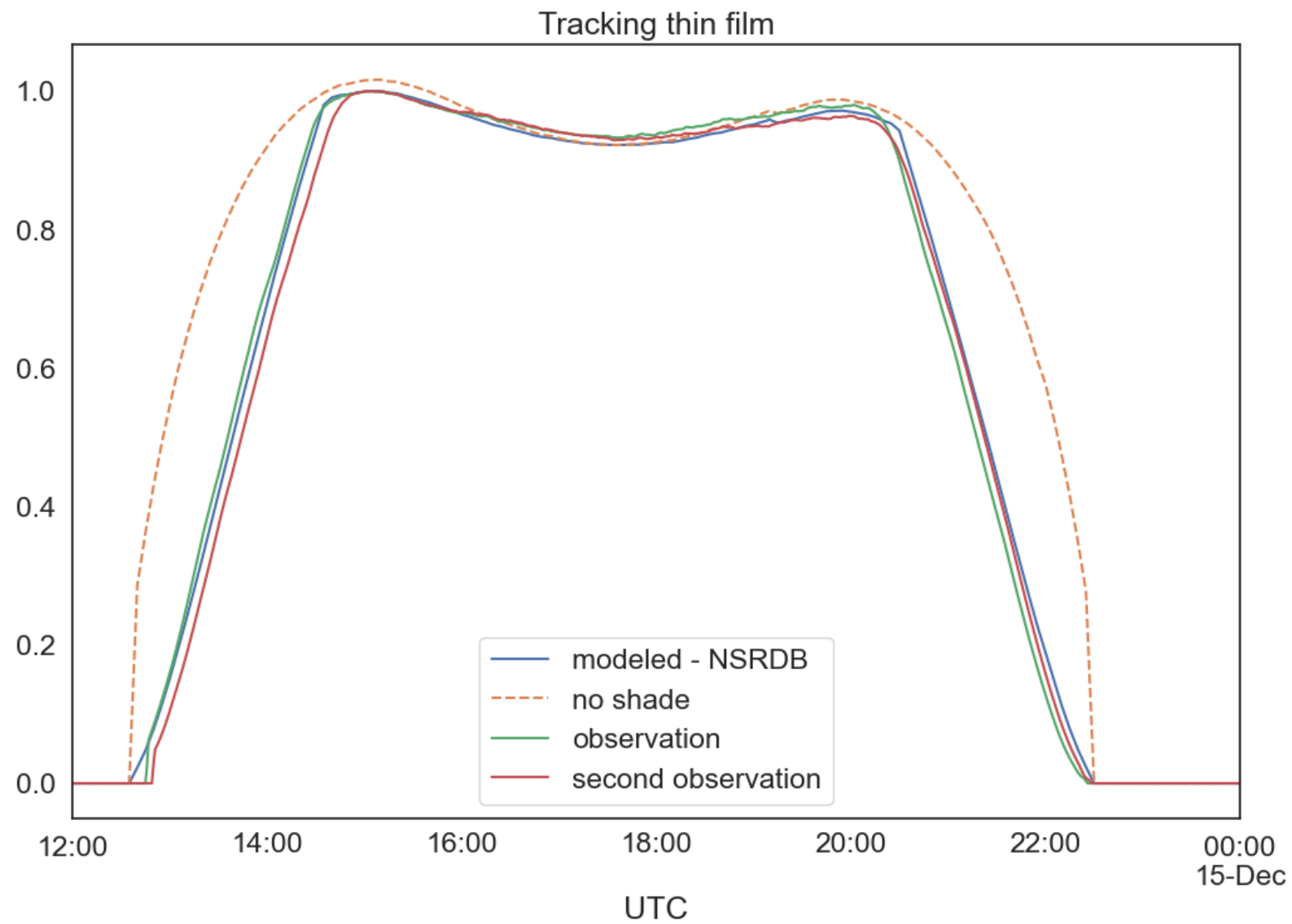


Some results

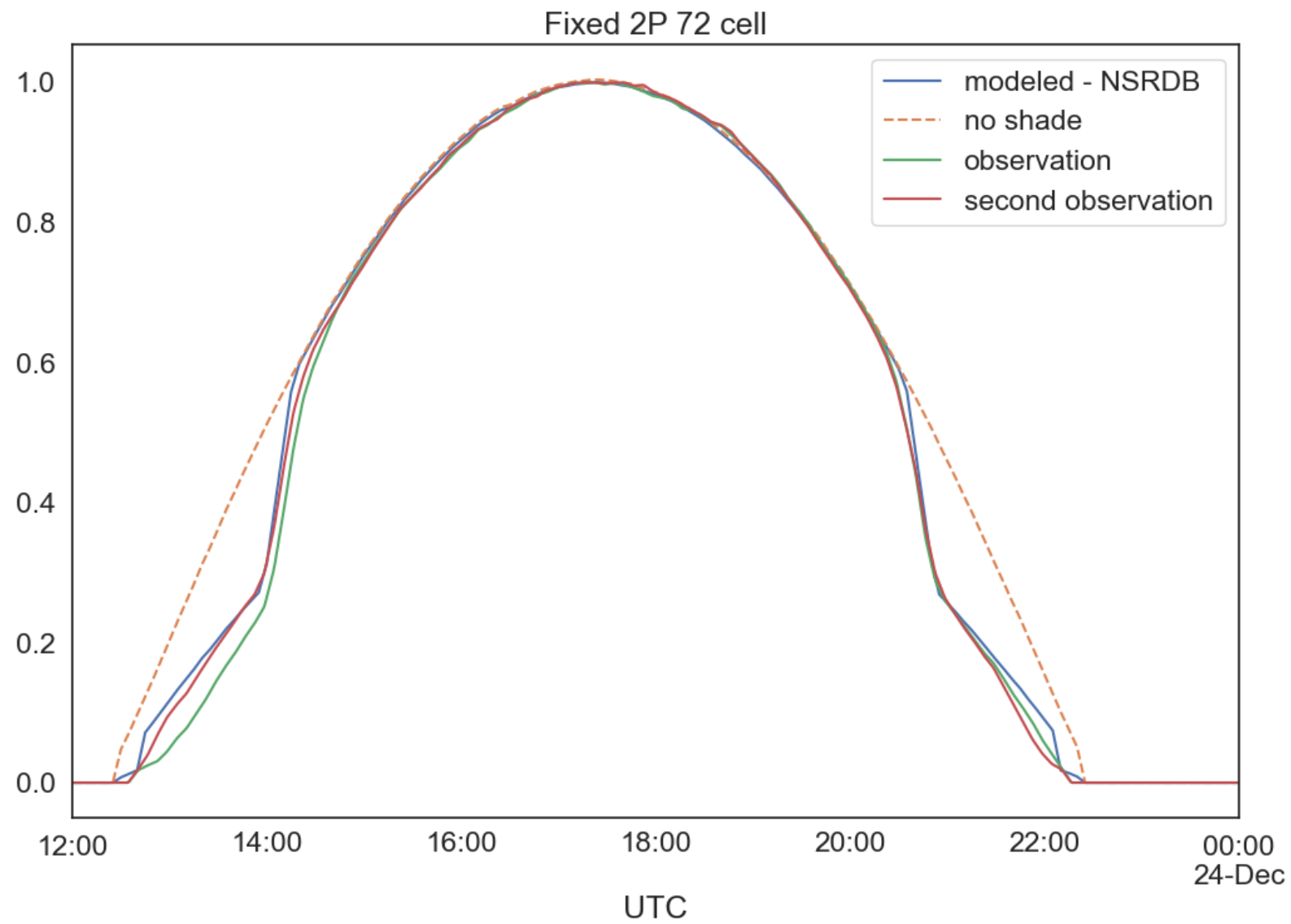
with observations from 2
inverters per plant



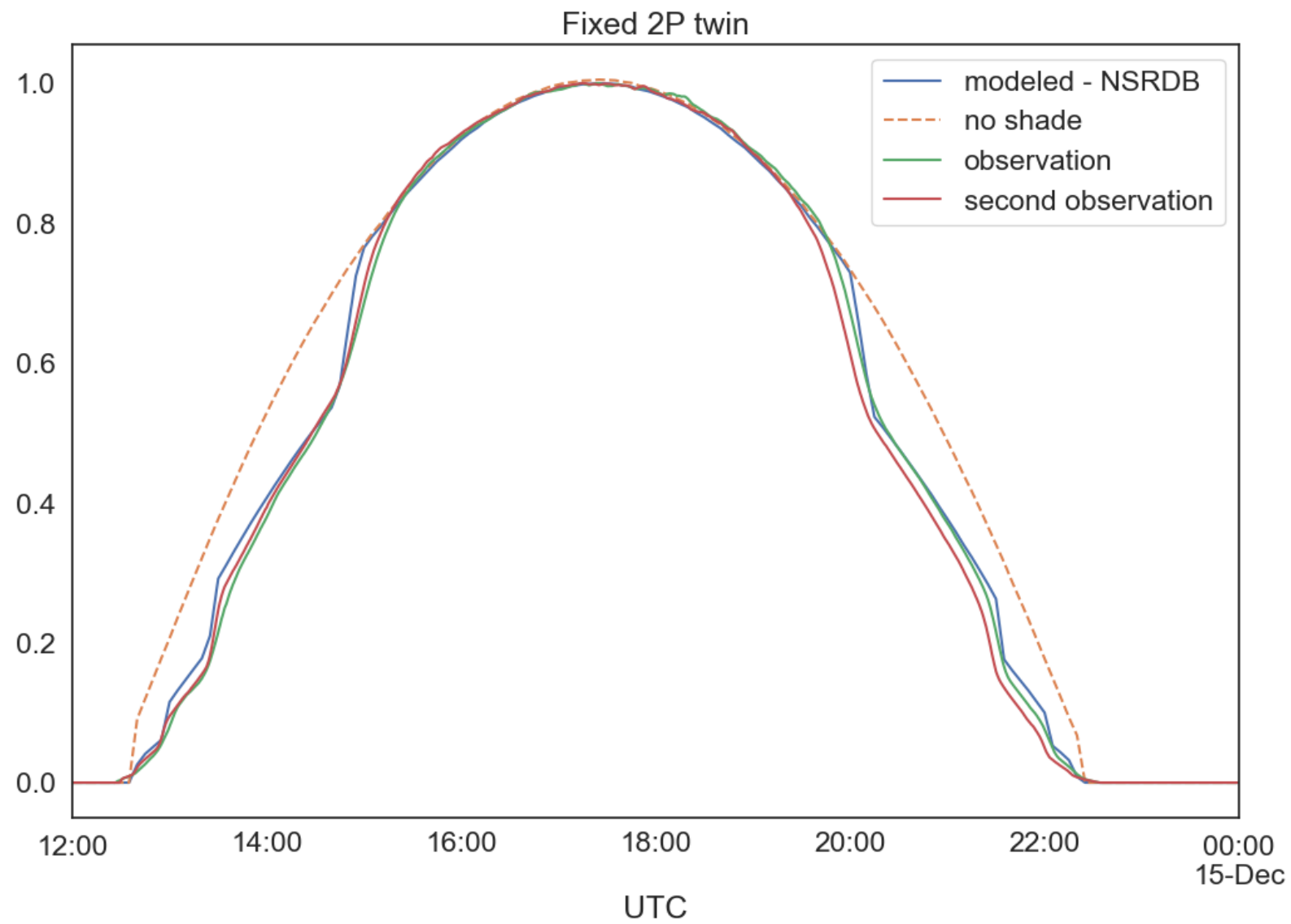
1.6% annual
difference (NSRDB)



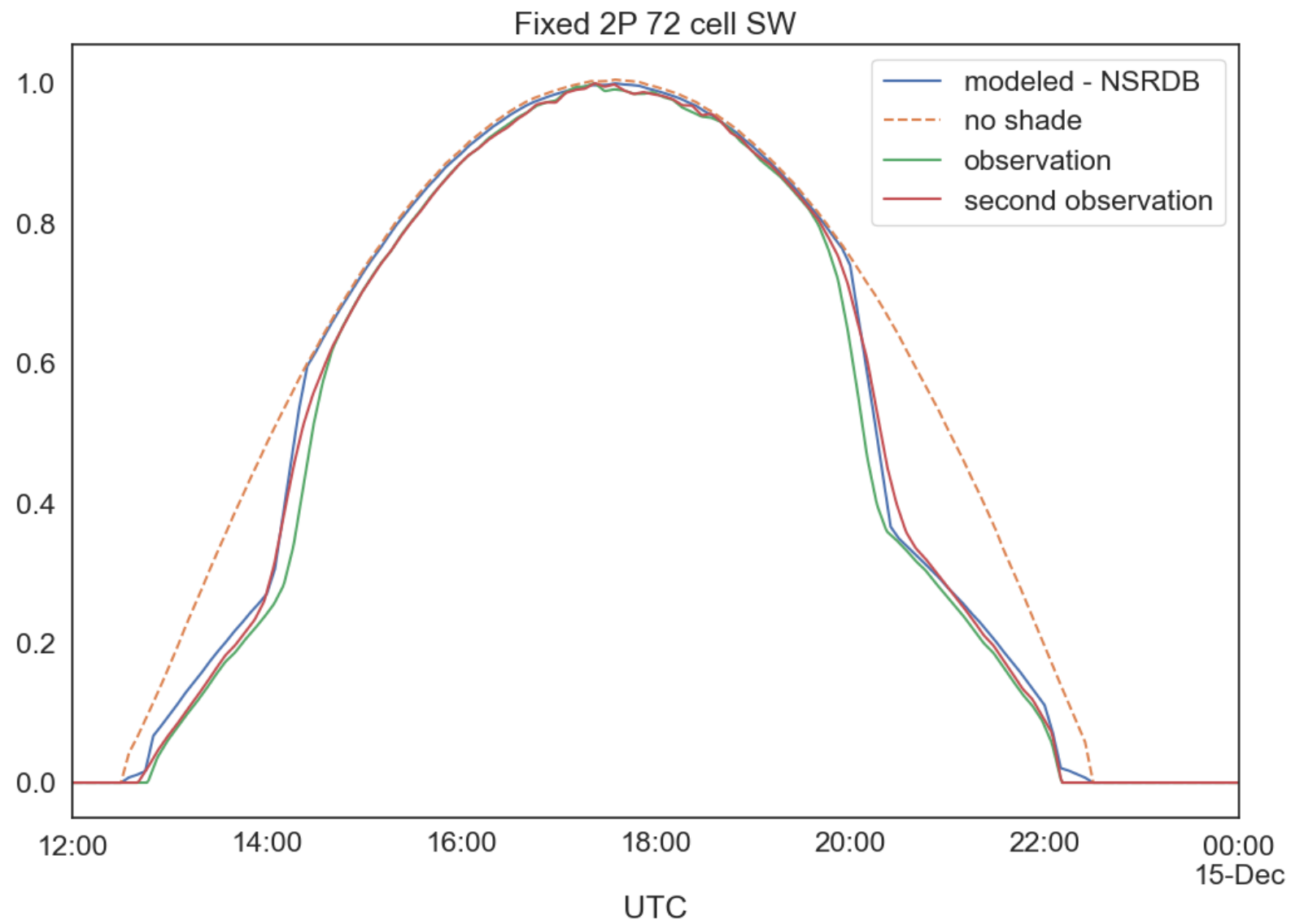
8.5% annual
difference (NSRDB)



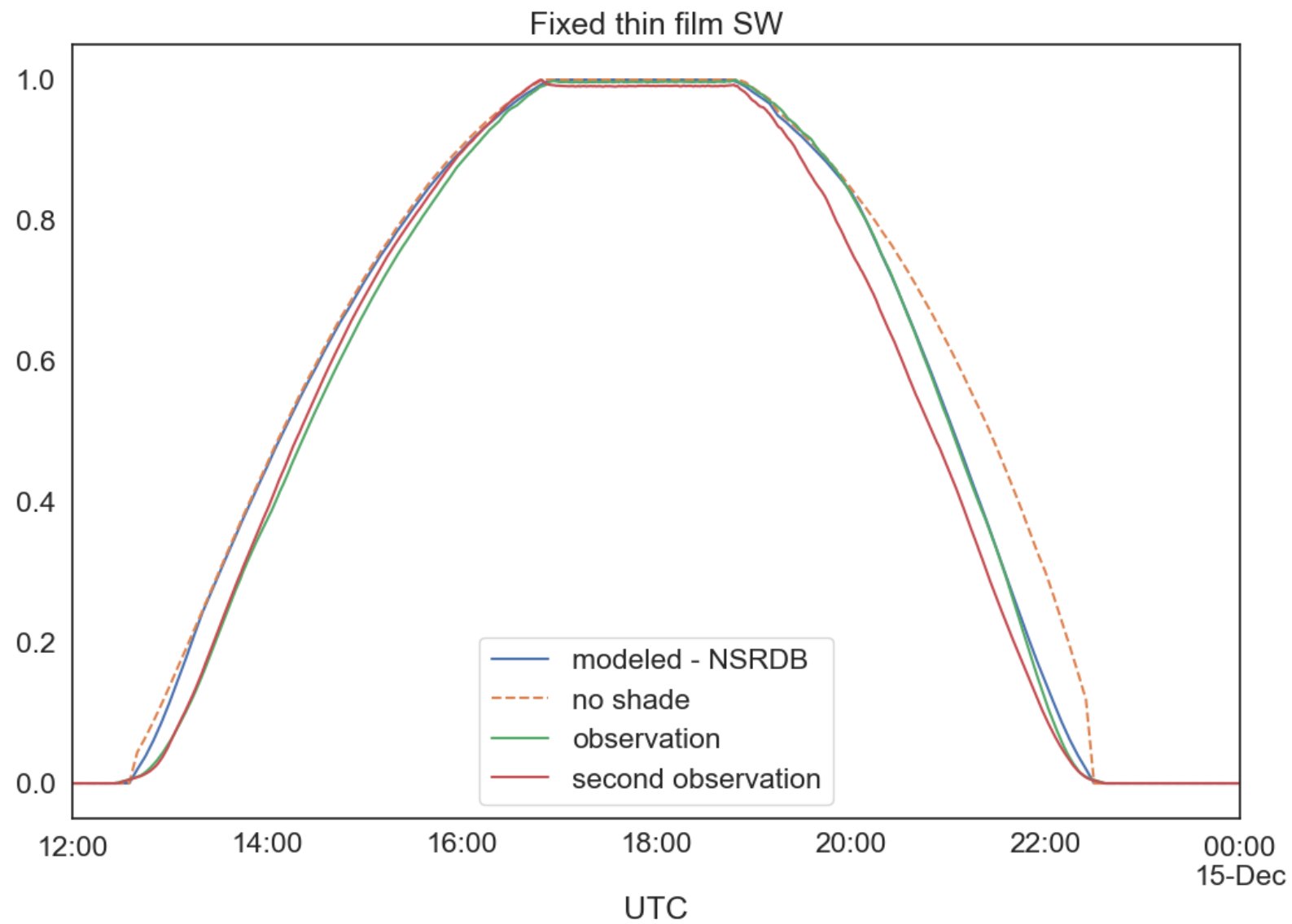
1.9% annual
difference (NSRDB)



2.5% annual
difference (NSRDB)

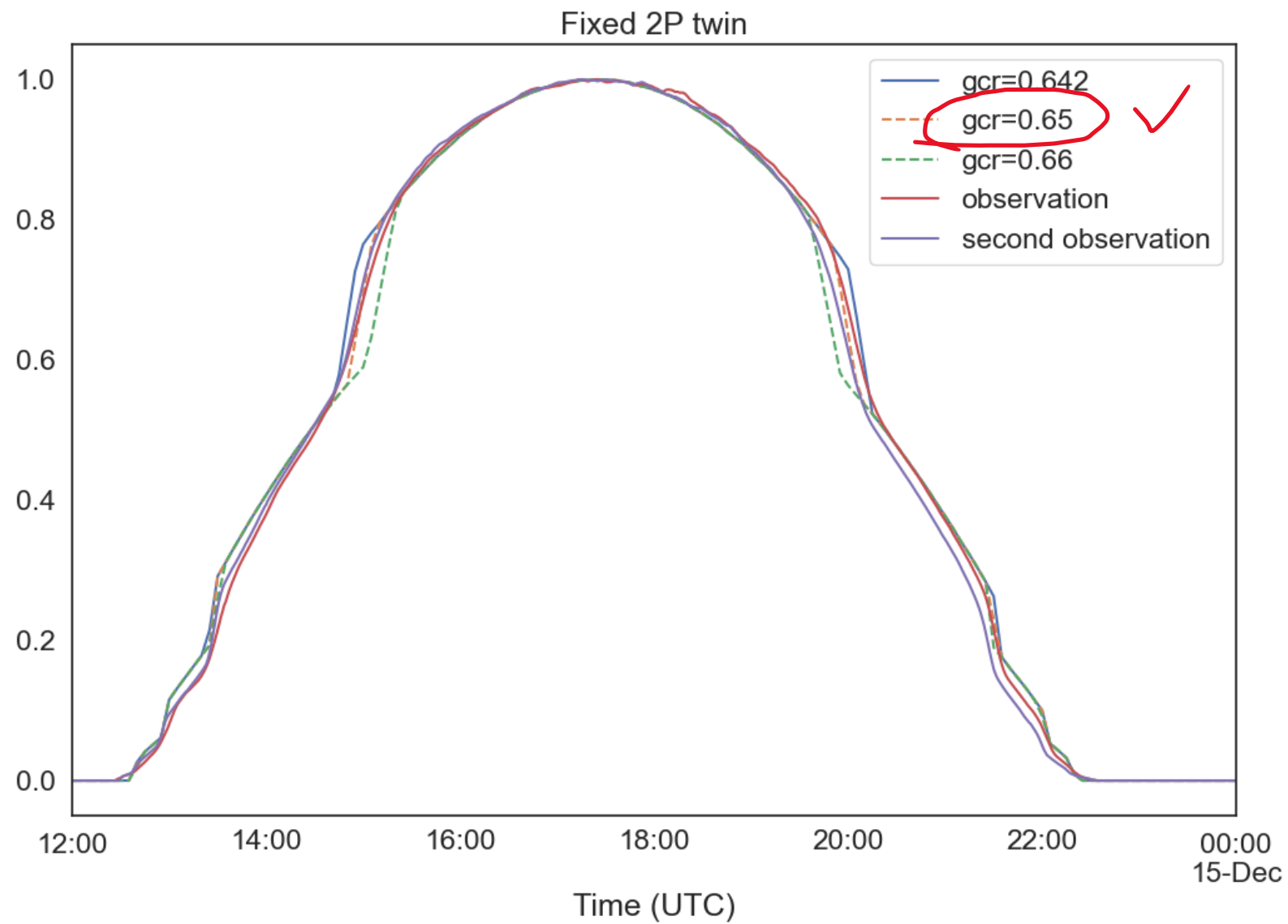


2.7% annual
difference (NSRDB)

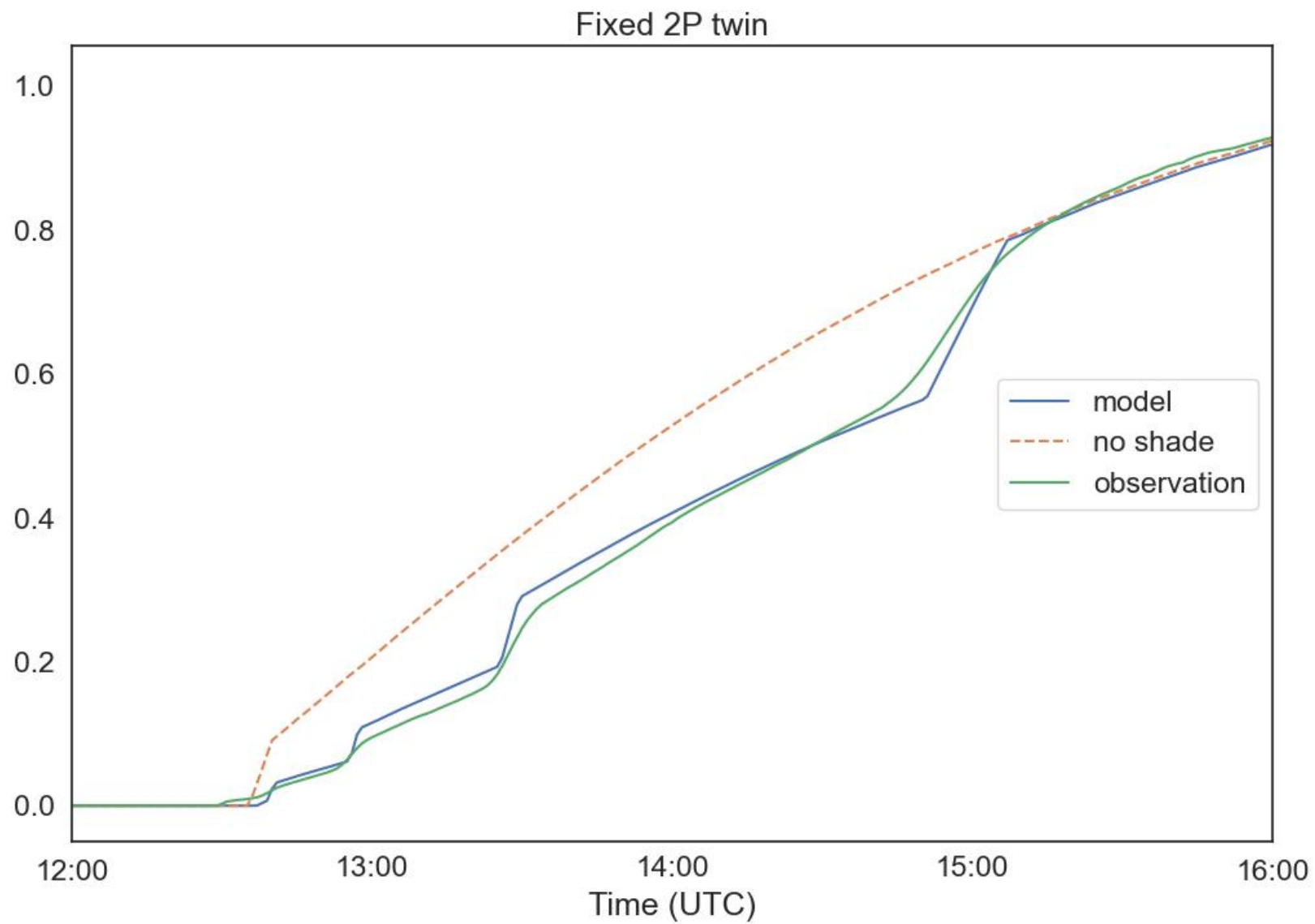


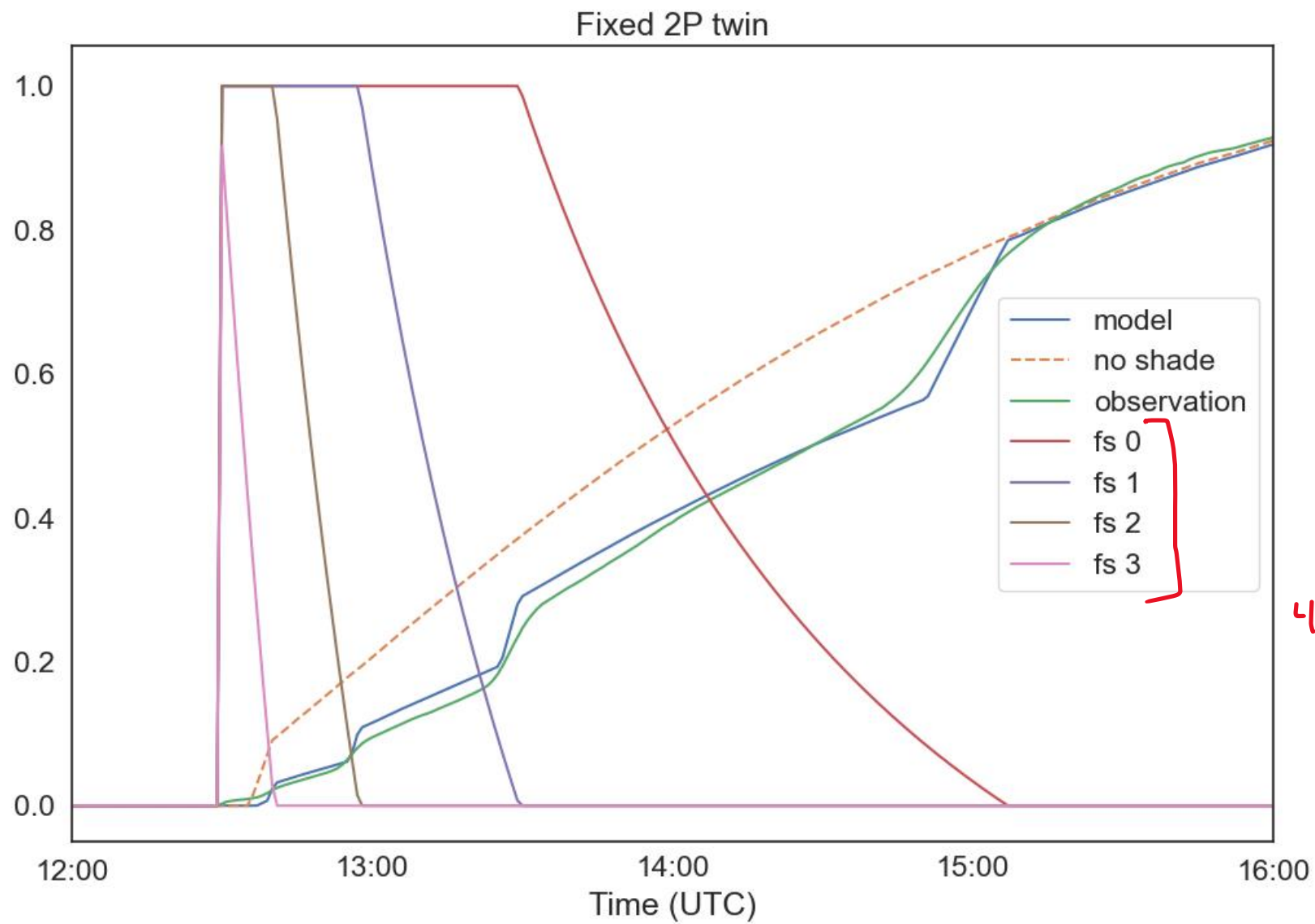
2.3% annual
difference (NSRDB)

Back to the 2P twin plant
let's look at GCR

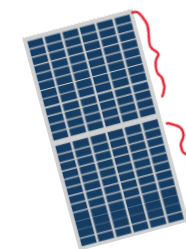


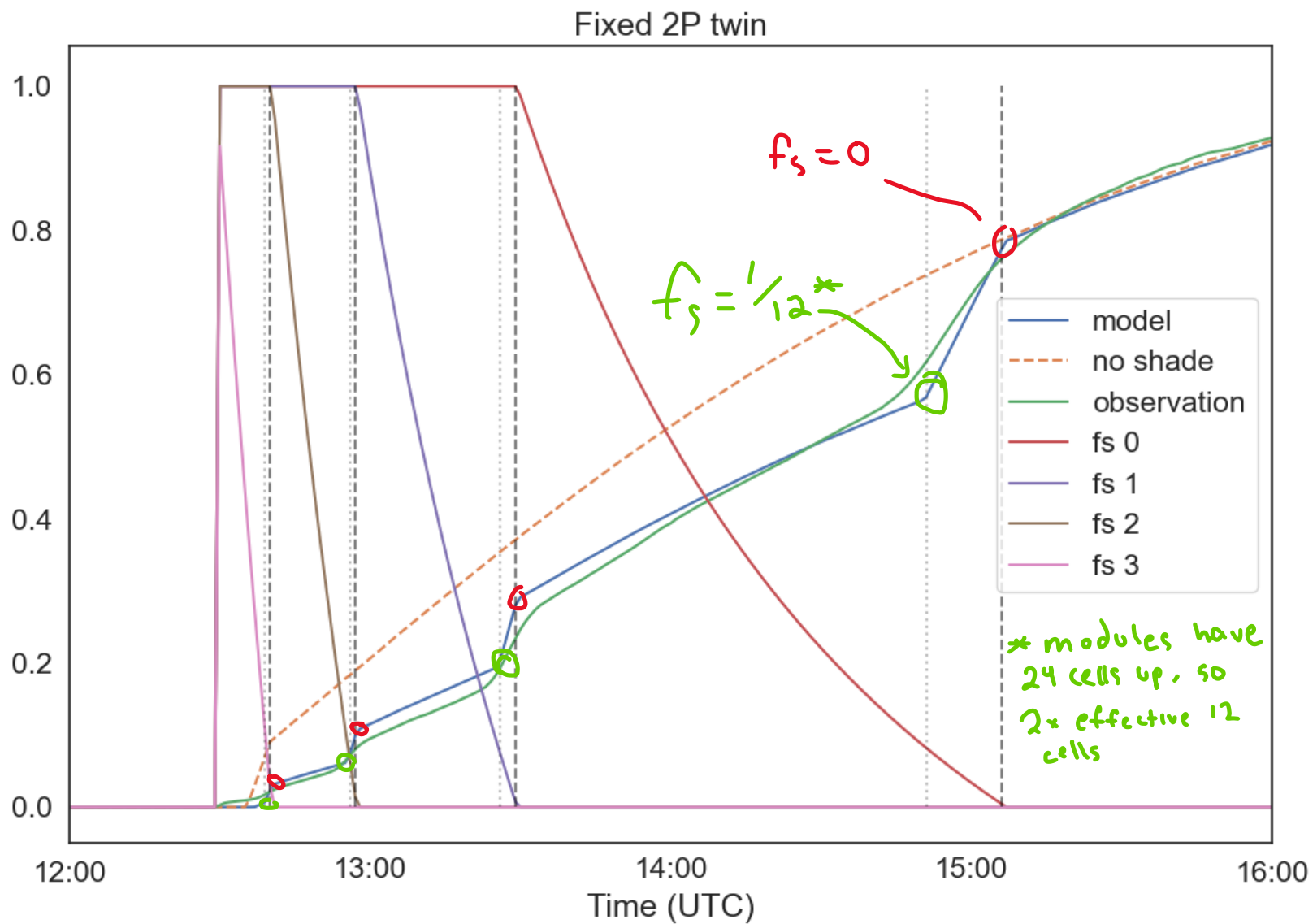
Let's look at the steps
zoom in on the morning





4 effective
courses
b/c twin



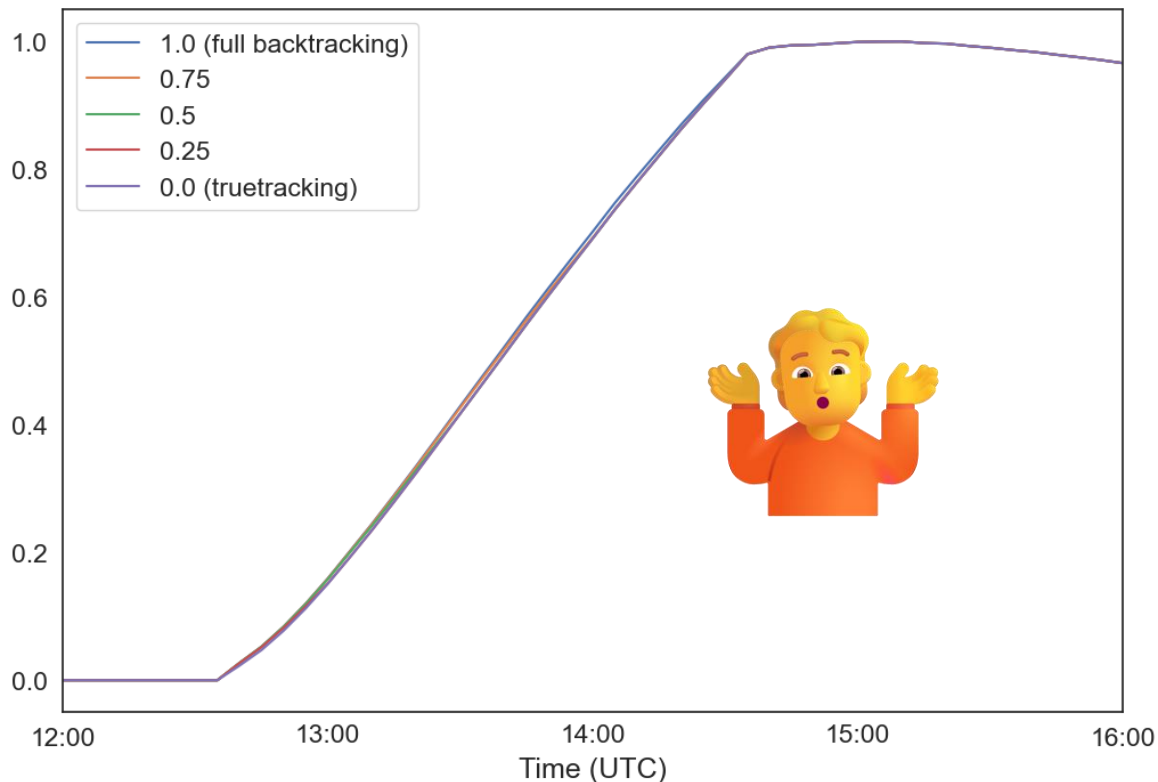


Other applications

fractional backtracking,
backtracking faults

Fractional Backtracking

- For twin modules, especially 2P (effectively 4P conventional) backtracking to allow shade for a fraction of the array might make sense (maybe even truetracking)
- Let's model a 2P twin example:

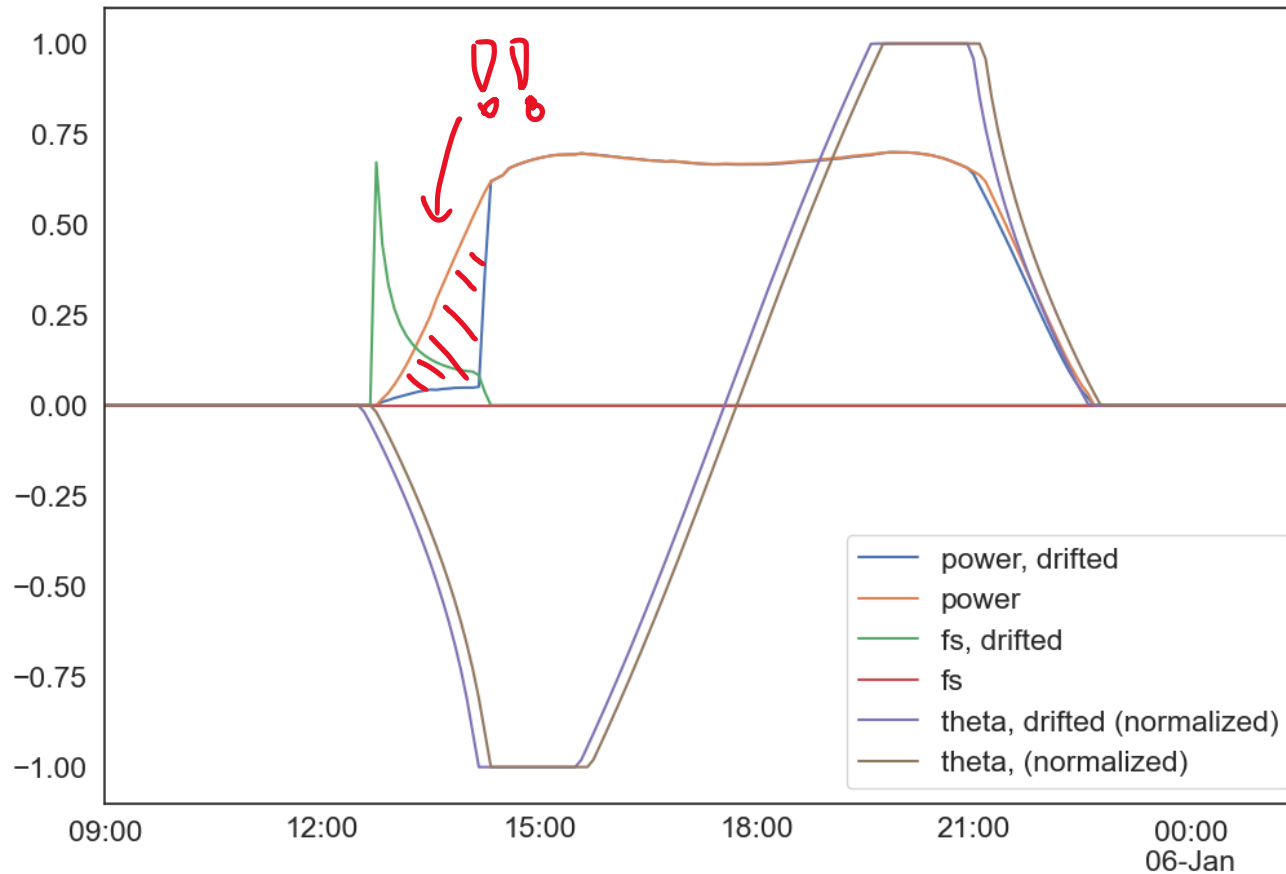


Mode	Annual Energy kWh (NSRDB)
✓ Full backtracking (1.0)	2058
0.75 backtracking	2050
0.5 backtracking	2046
0.25 backtracking	2044
0.0 (truetracking)	2043

Resource-dependent??

Misaligned trackers (clock drift)

- Tracker clock is 10 minutes fast
- 1P 72-cell crystalline



**~5% annual
losses!**
(NSRDB)

Code availability
it's open-source!

https://github.com/williamhobbs/2024_pvpmc_self_shade

Repo includes everything shown here:

- Functions
- example notebooks with plots
- anonymized observation data and specs



Functions

- `shade_fractions(fs_array, eff_row_side_num_mods)`

- `non_linear_shade(n_cells_up, fs, fd)`

- `plant_power_with_shade_losses(`

`resource_data,`

`**plant_data,`

`surface_tilt_timeseries,`

`surface_azimuth_timeseries,`

`use_measured_poa,`

`use_measured_temp_module)`

Dictionary of specs

(<https://github.com/williamhobbs/pv-plant-specifications>)

Optional

INCLUDES BIFACIAL!!

Questions?

whobbs@southernco.com



[**https://github.com/williamhobbs/2024_pvpmc_self_shade**](https://github.com/williamhobbs/2024_pvpmc_self_shade)



Room for improvement

- Account for “inactive bands” at module edges (or, at least, see how much it matters)
- Crystalline modules in landscape
- Unconventional crystalline module designs (shingled cells, etc.)
- Comparison with single diode model
- Slopping terrain (<https://github.com/pvlib/pvlib-python/pull/1962> by echedey-ls will help)