THE UNIVERSITY OF QUEENSLAND
AUSTRALIA
CREATE CHANGE

Dynamic operating envelopes

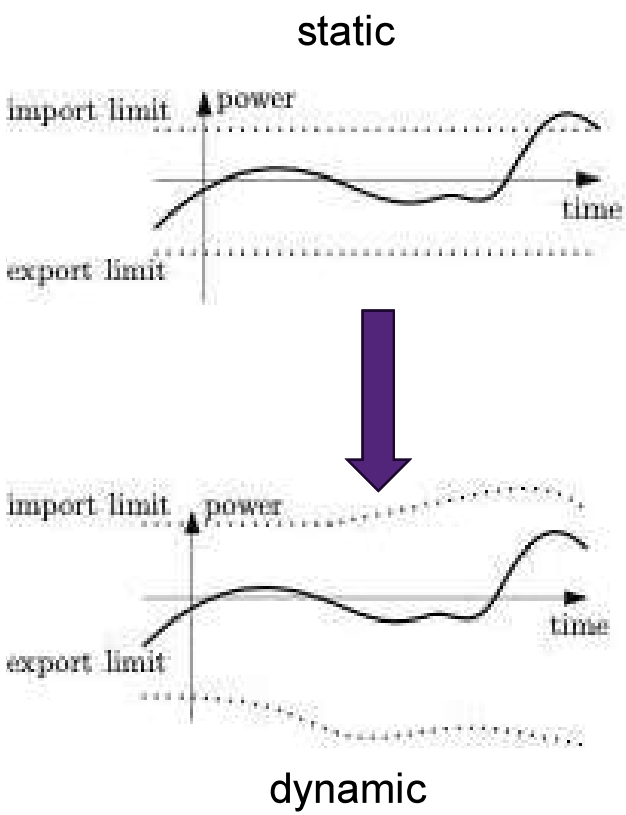
From static exports to DOEs

In Australia, you are typically allowed to *install* rooftop PV up to 5 kW per phase

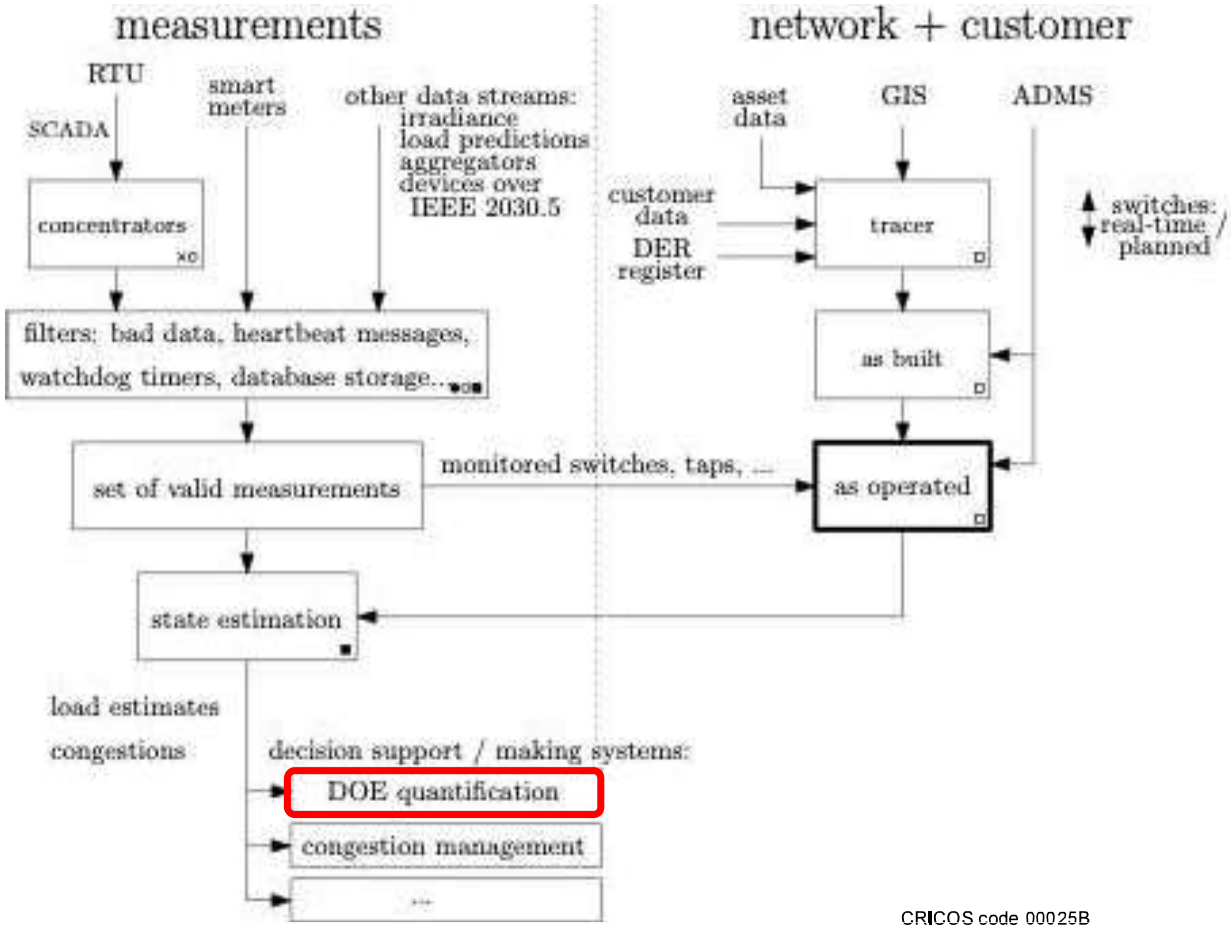
- However, you may *not* be allowed to *export* power to the network, e.g. you can only inject 1.5 kW, or even 0.
- These static limits are driven by voltage-based congestion

In practice, even in congested networks, there is spare capacity a lot of the time, which is not accessible under these static export limits

… make them dynamic?



Visibility to enable DOE



Operating envelopes

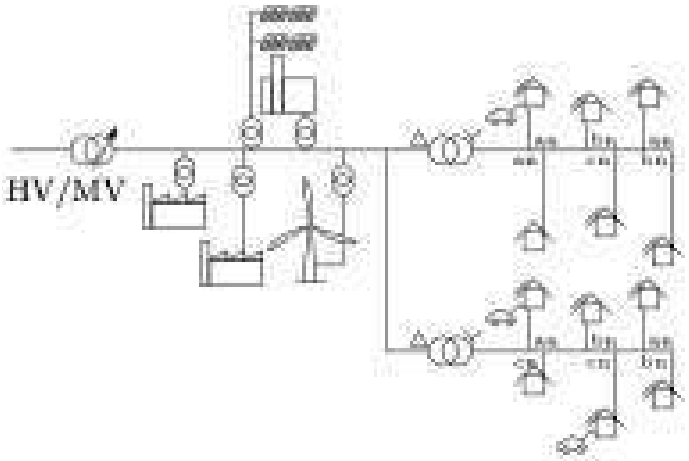
Maximum export values that can be realized simultaneously without violating the network envelopes for voltage magnitude and thermal limits

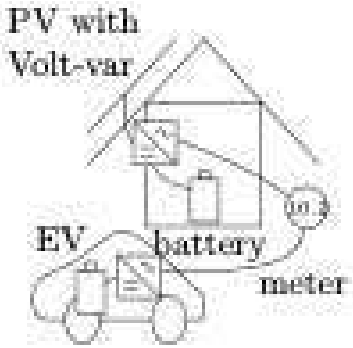
- Optimisation is a natural framework for finding minima and maxima

Approach: optimisation problems that include the distribution network physics

- Including expected inverter response
- Taking into account different objectives for equity and competitiveness

DOE





OPF example

min. cost of dispatch
s.t. network physics
operating limits: S, V, I
devices: EV, PV, battery
inverter control

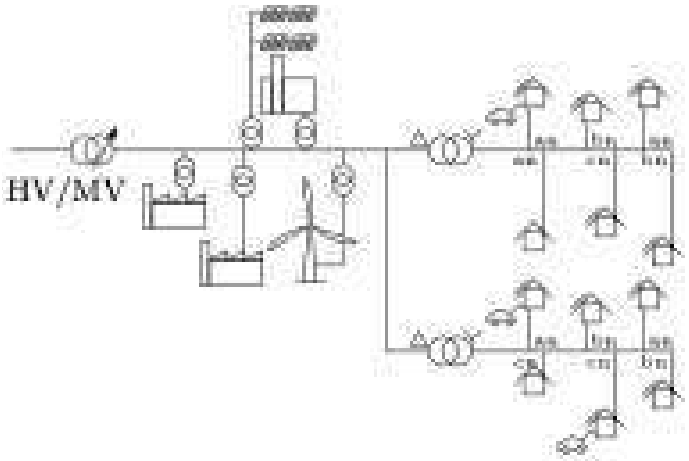
outcome:
feasible dispatch &
network within
operating limits

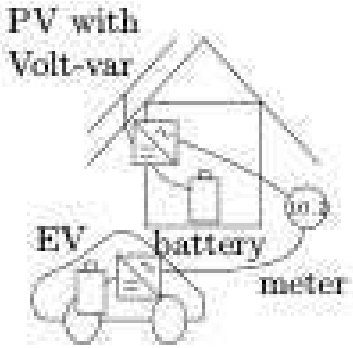
DOE example

max. aggregate exports
s.t. network physics
operating limits: S, V, I
devices: EV, PV, battery
inverter control

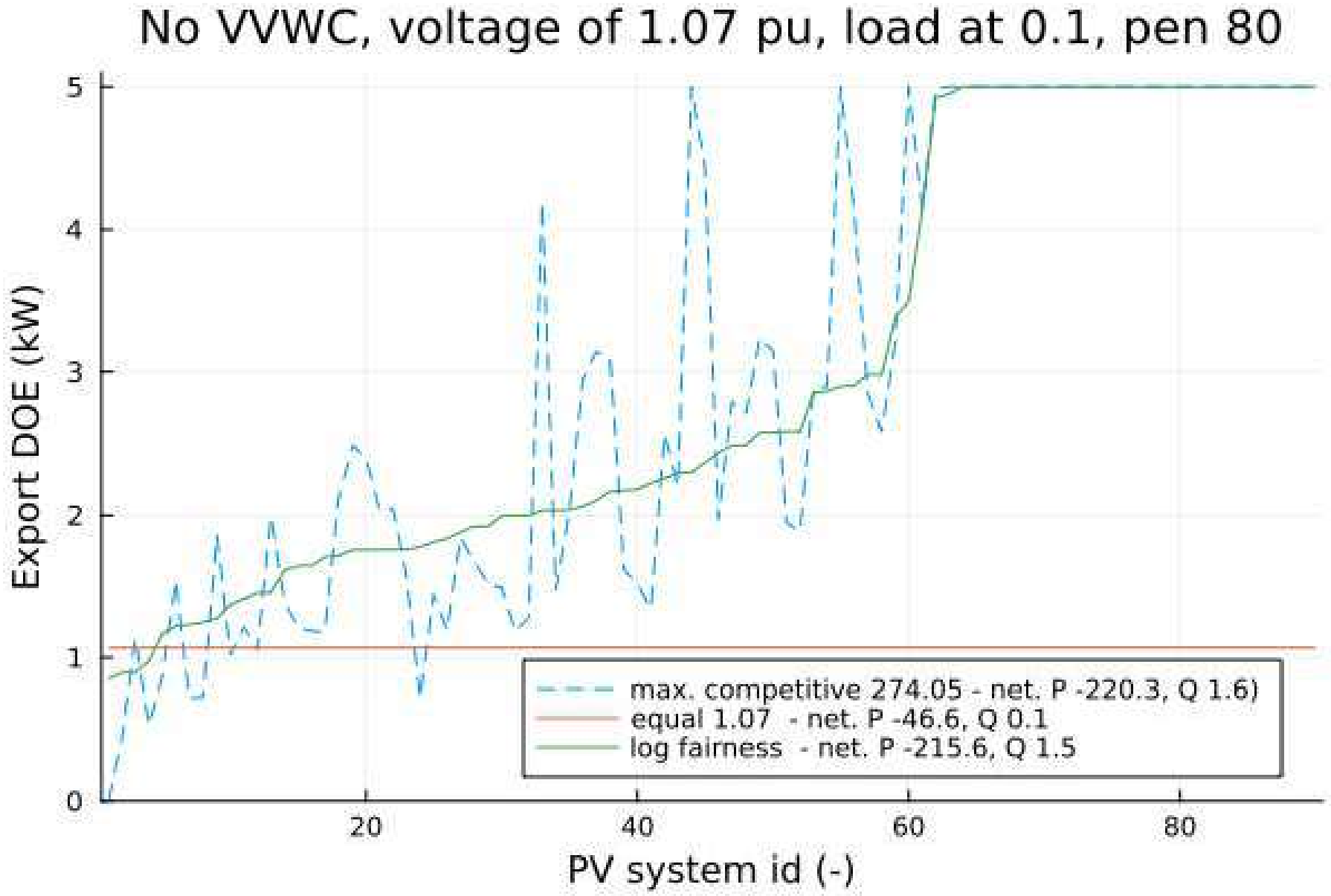
outcome:
feasible export ranges

DOE

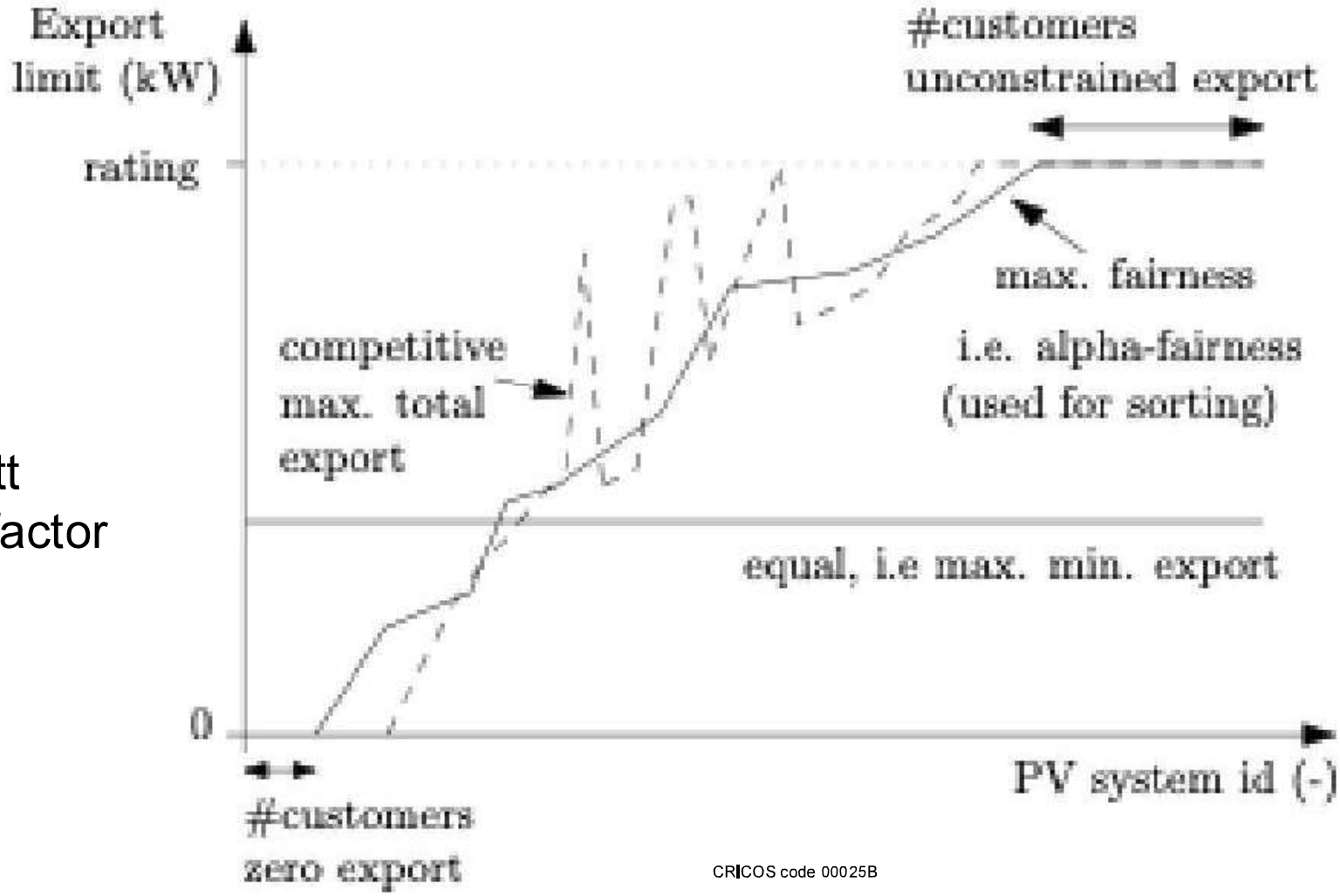




OPF example	DOE example
min. cost of dispatch	max. aggregate exports
s.t. network physics	s.t. network physics
operating limits: S, V, I	operating limits: S, V, I
devices: EV, PV, battery	devices: EV, PV, battery
inverter control	inverter control
outcome:	outcome:
feasible dispatch & network within operating limits	feasible export ranges

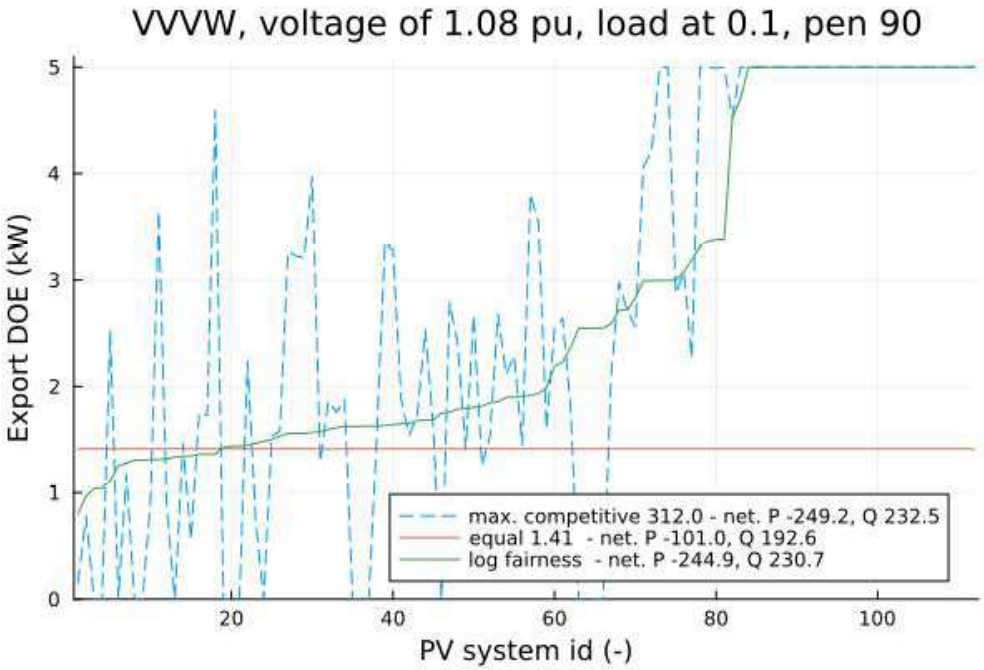

No VVWC, voltage of 1.07 pu, load at 0.1, pen 80
Export DOE (kW)
PV system id (-)
0
1
2
3
4
5
20
40
60
80
max. competitive 274.05 - net. P -220.3, Q 1.6)
equal 1.07 - net. P -46.6, Q 0.1
log fairness - net. P -215.6, Q 1.5

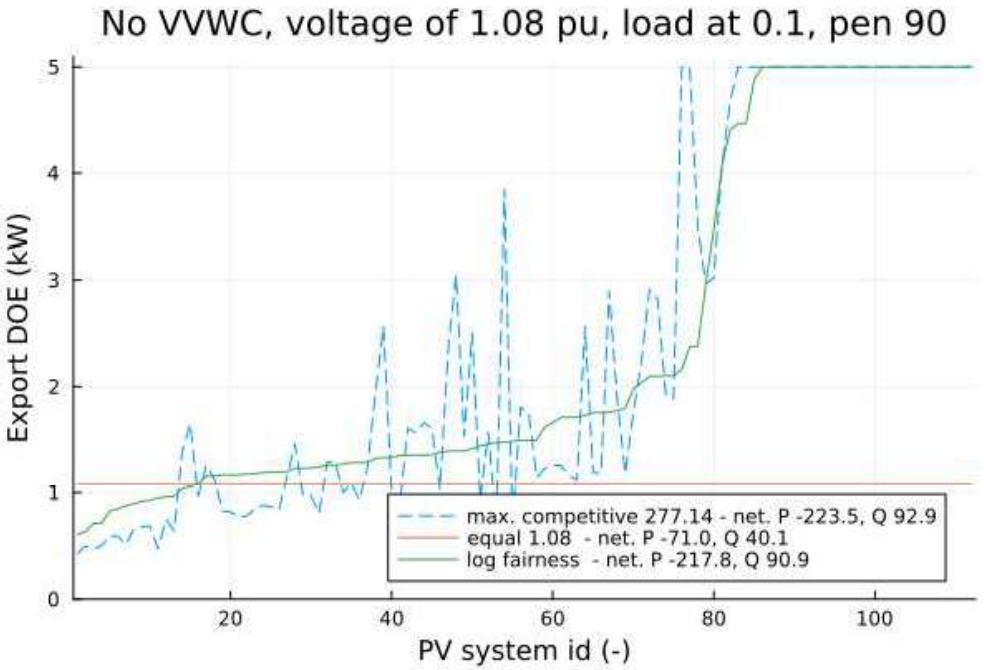
Studies across
- PV penetration
- Load levels
- Voltage at infeeder
- PV with volt-var/watt
  vs constant power factor



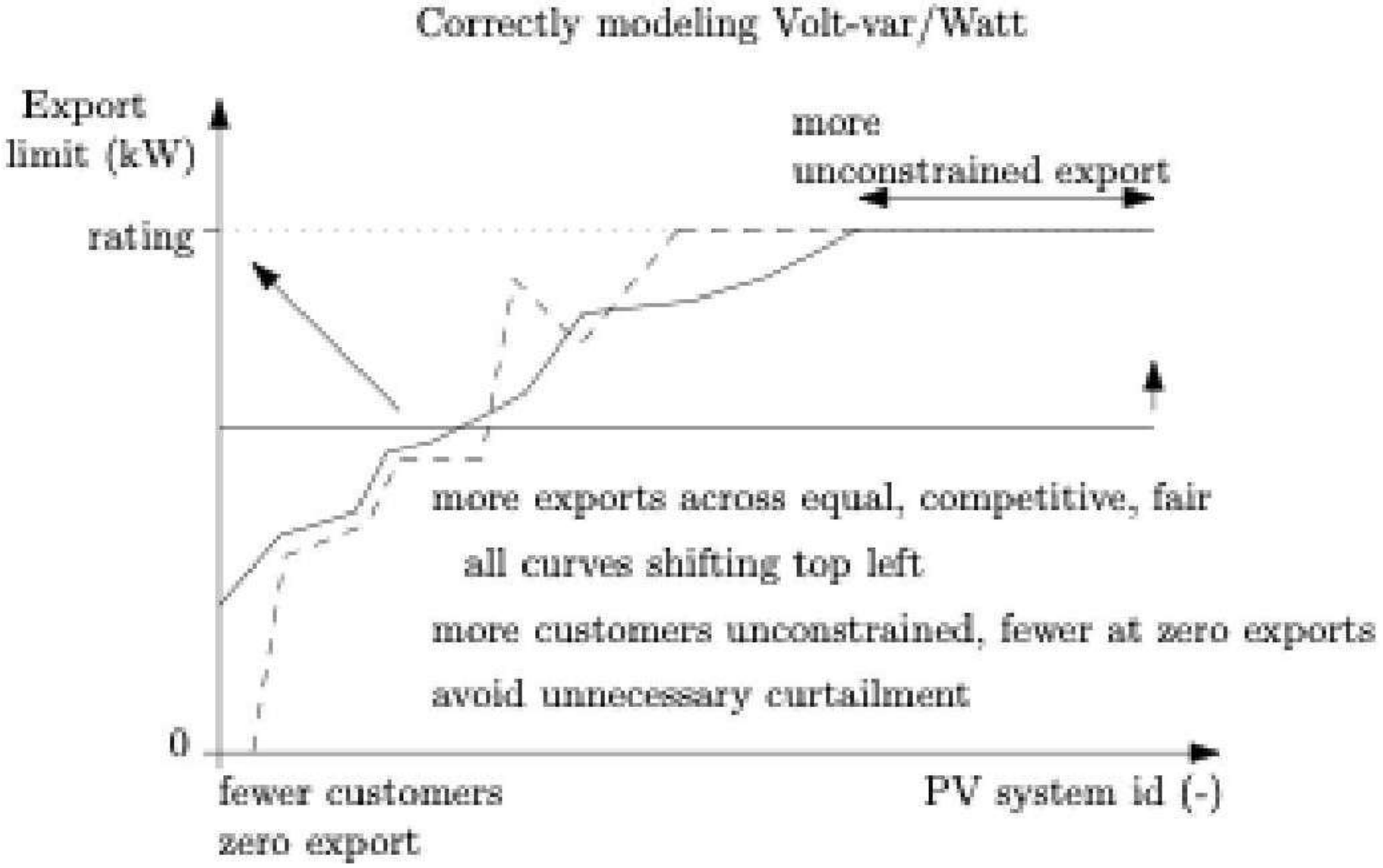
CRICOS code 00025B

With vs without volt-var/watt





CRICOS code 00025B


Correctly modeling Volt-var/Watt
Export
limit (kW)
rating
more
unconstrained export
more exports across equal, competitive, fair
all curves shifting top left
more customers unconstrained, fewer at zero exports
avoid unnecessary curtailment
0
fewer customers
zero export
PV system id (-)
CRICOS code 00025B

Challenges

How to make the choice of a specific DOE fairness trade-off as utility or policy maker

Flexible import limits: flexible vs inflexible demand

- EVs, battery, pool heaters

Incorporating OLTCs and statcoms into DOE quantification

Network planning methods taking the presence of DOEs into account

Network data debugging, cleaning, calibration, validation methods

How to measure and validate the outcomes of DOE *deployments*

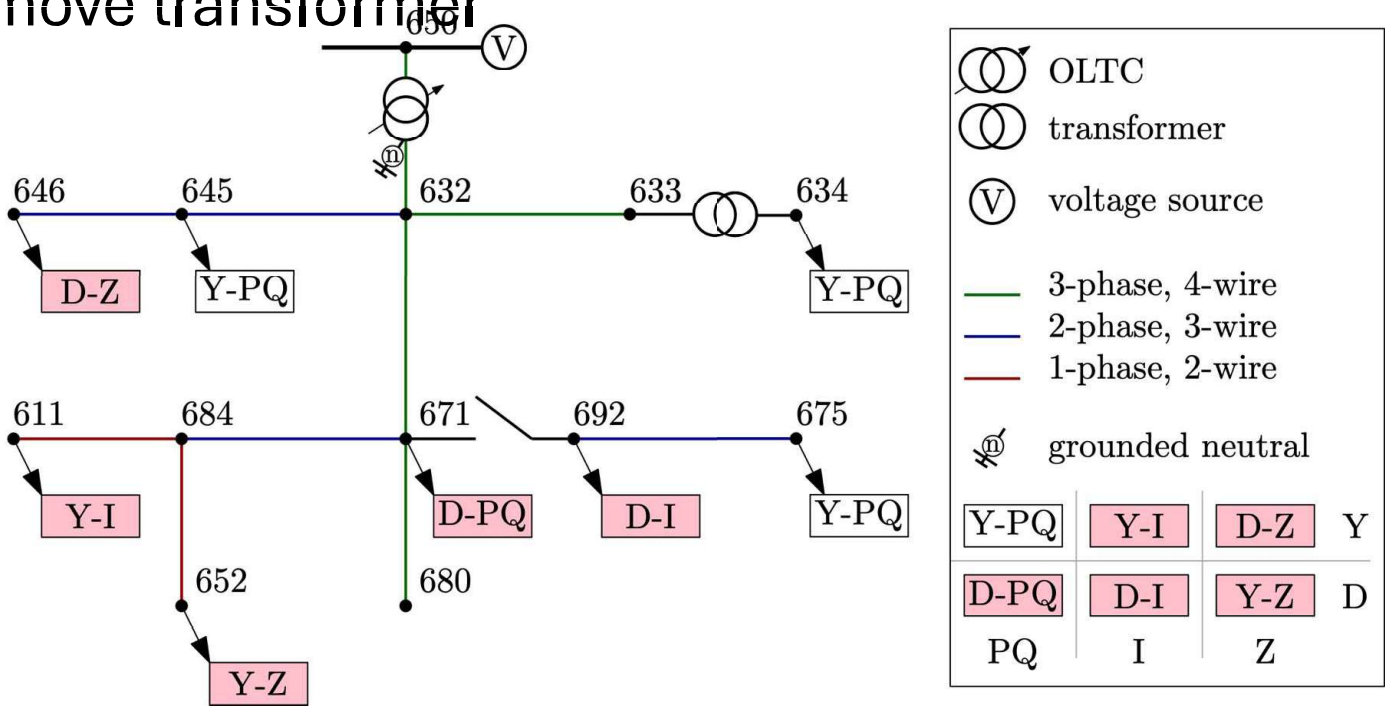
Flex markets and DOEs: compatibility and complementarity?

IEEE Distribution test feeders

- https://cmte.ieee.org/pes-testfeeders/resources/
- Made for simulation, not optimization
- Designed to confirm consistent results between different simulation engines, for a large variety of network topologies, transformer designs

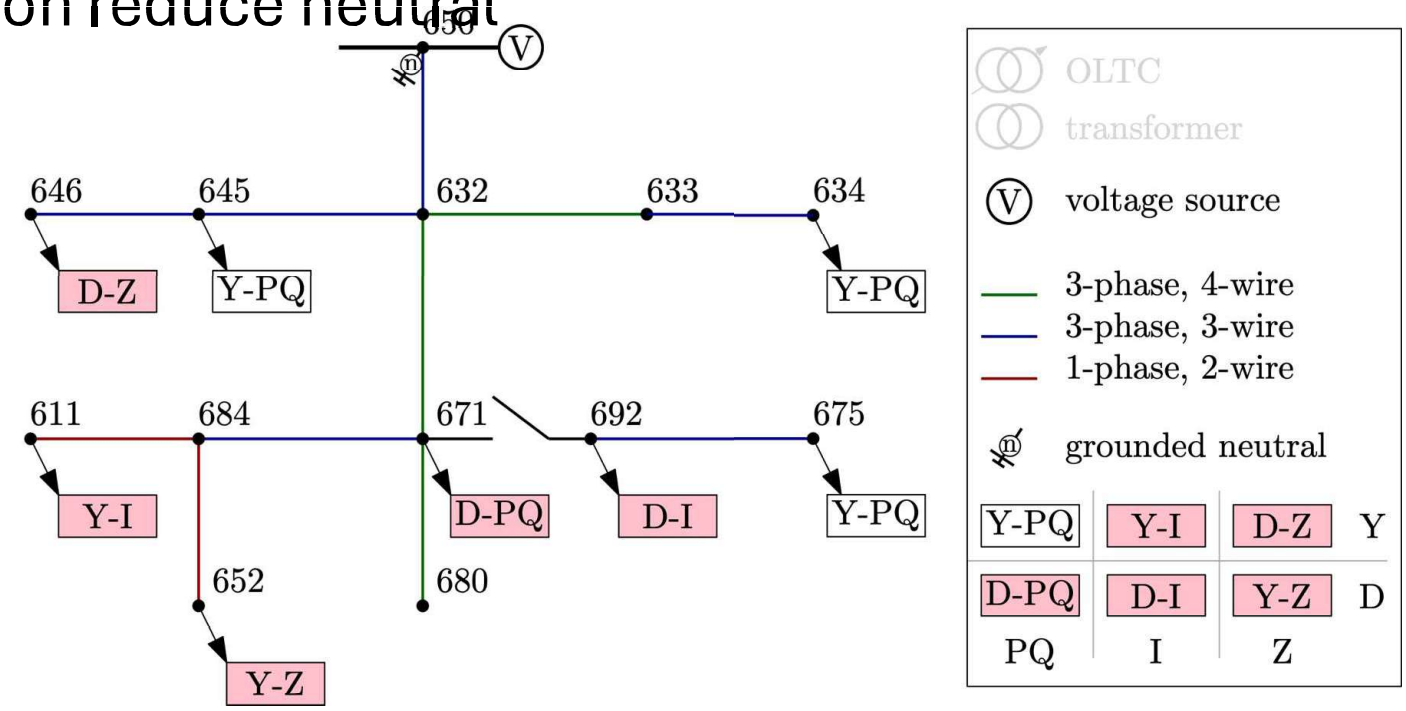
Example: IEEE 13

- remove transformer



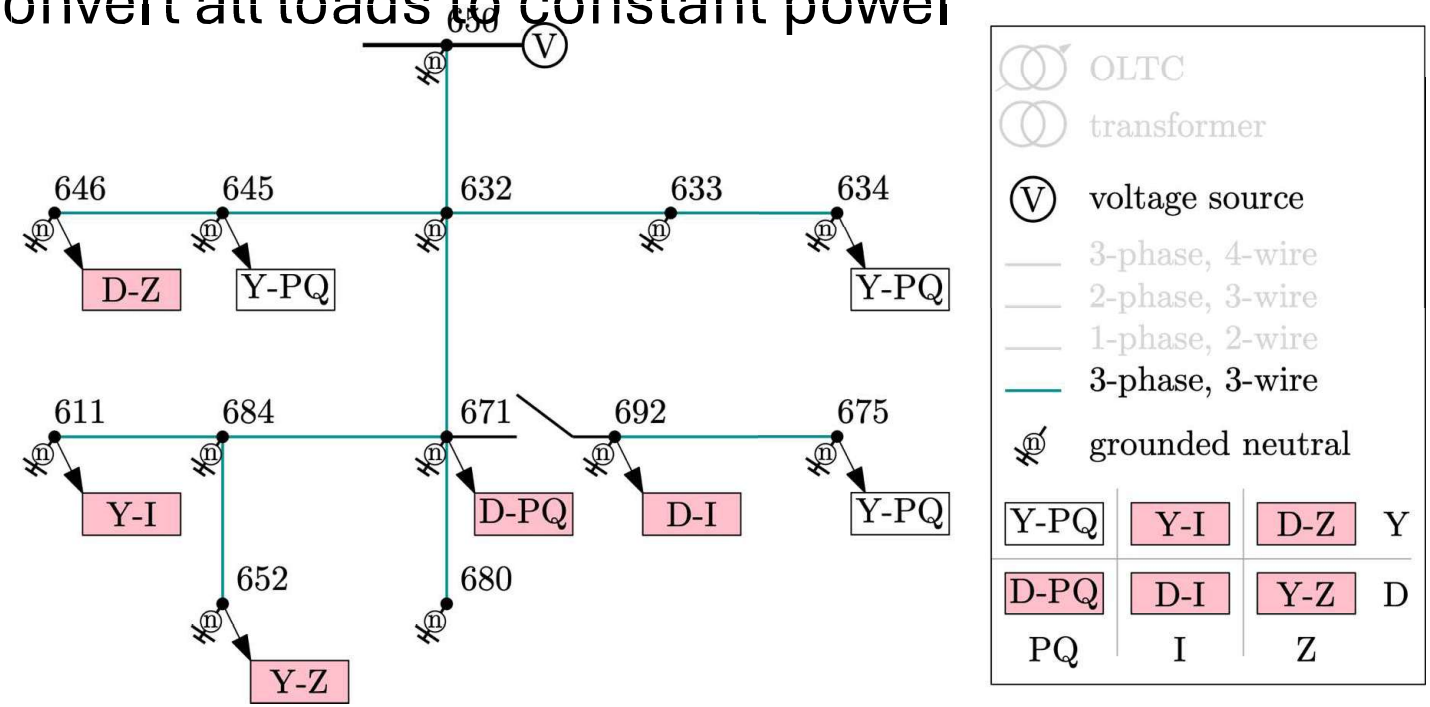
Example: IEEE 13

- Kron reduce neutral

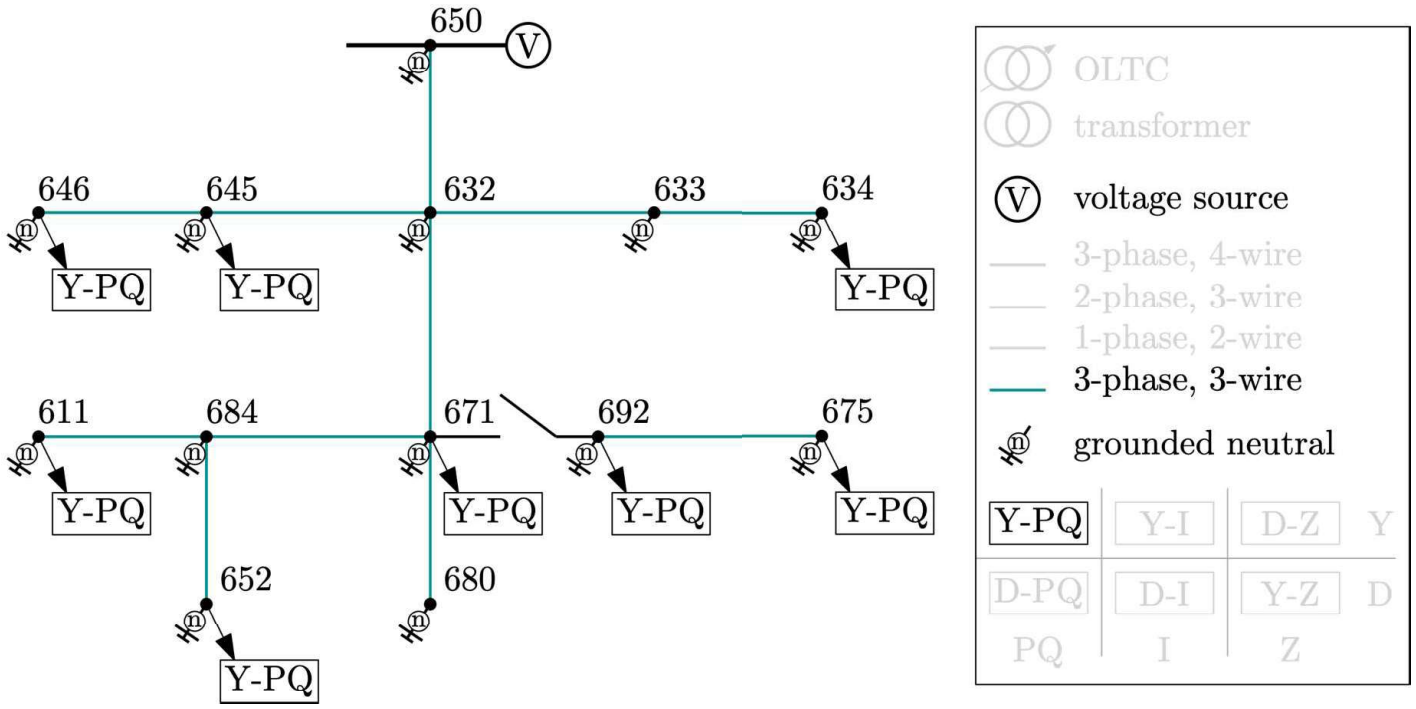


Example: IEEE 13

- convert all loads to constant power



Example: IEEE 13



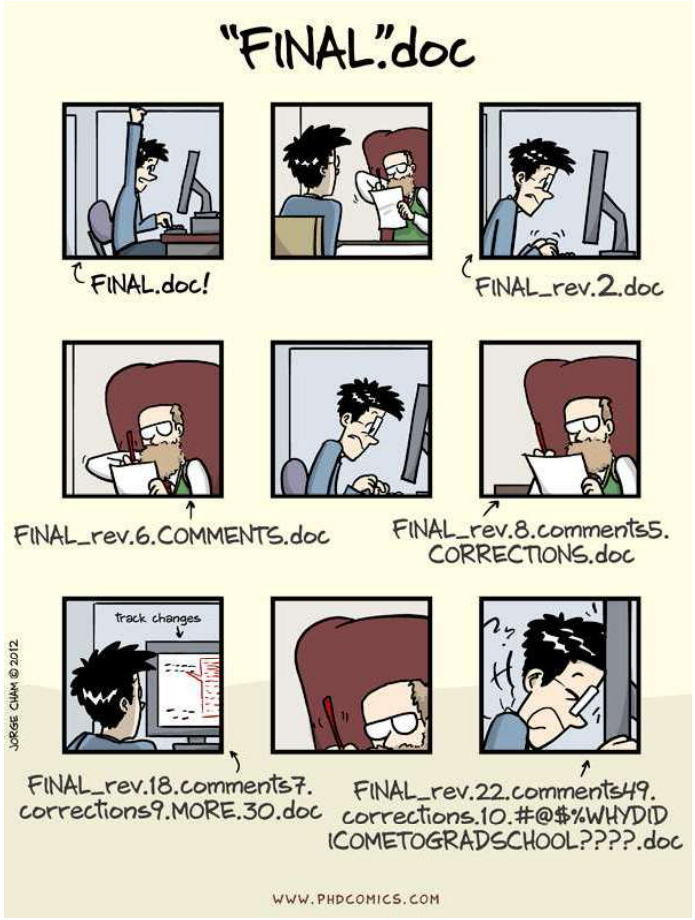
Australian Distribution Network Data

- https://data.csiro.au/collection/csiro:65408?q=geth%20heidari&_st=keyword&_str=3&_si=1

Parses in both PMD and OpenDSSDirect

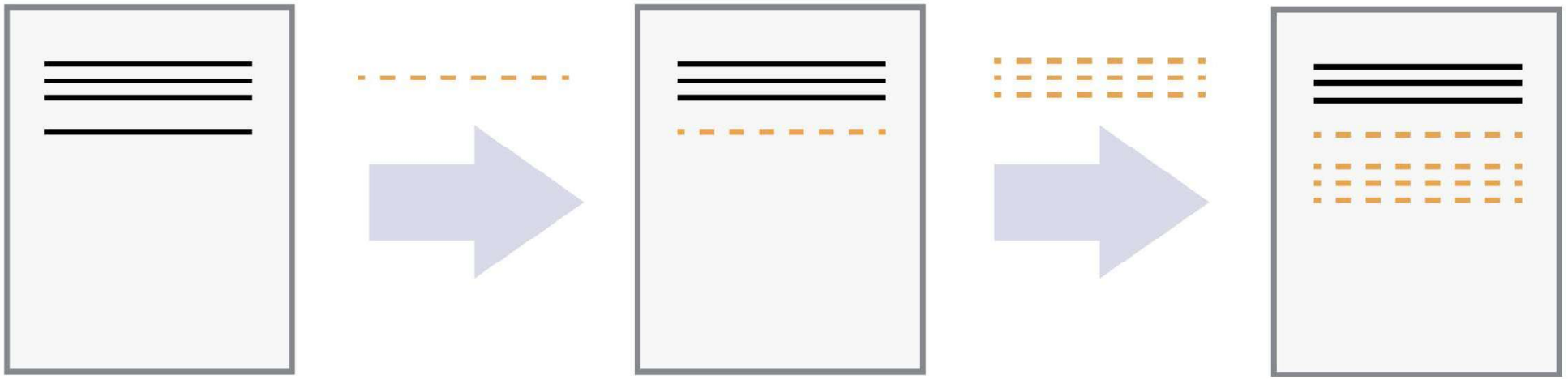
Automated Version Control

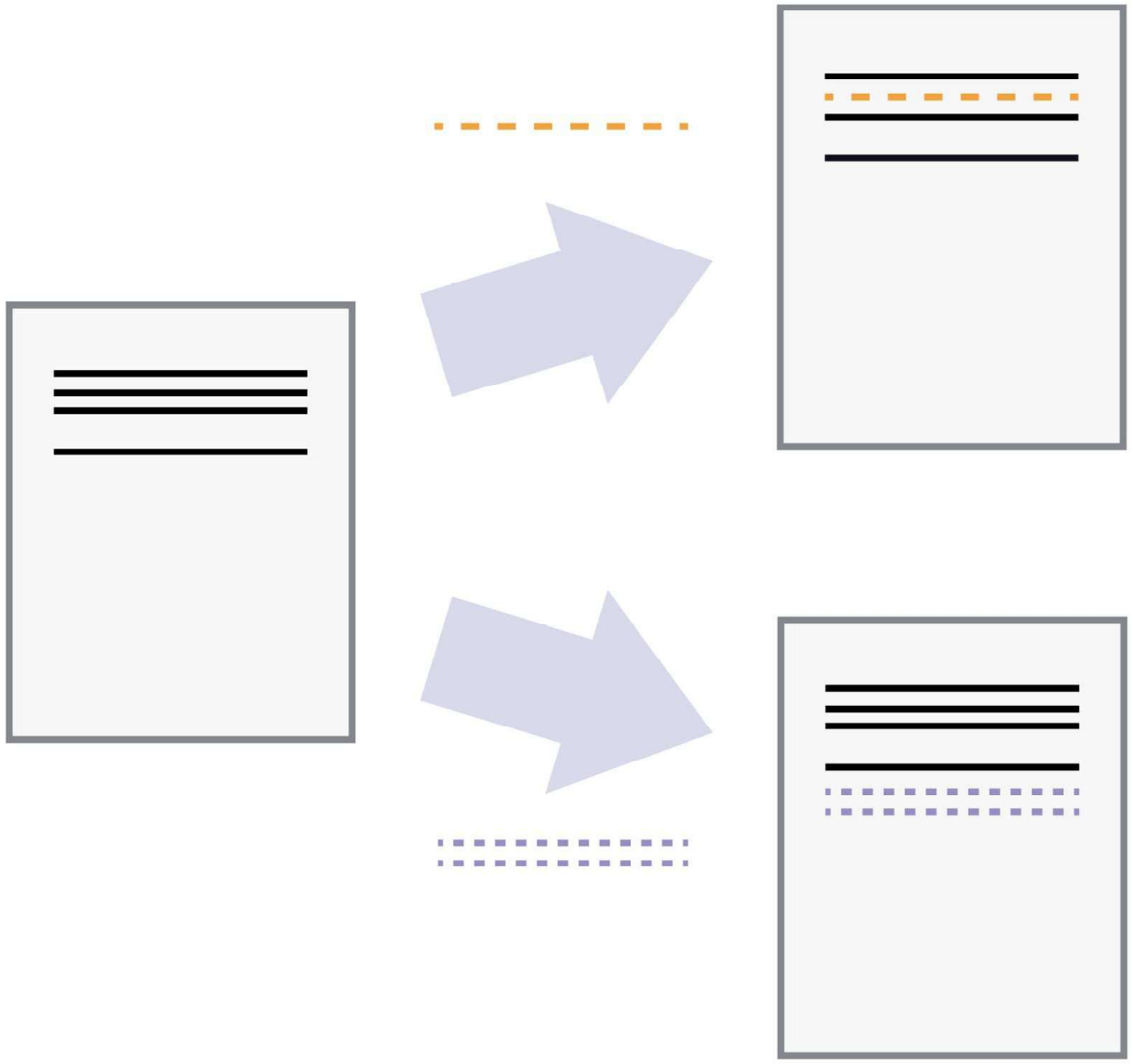
- Version control is like an unlimited ‘undo’.
- Version control also allows many people to work in parallel.
- Even if you work on your own, it can be very powerful
- Cheat code if you collab

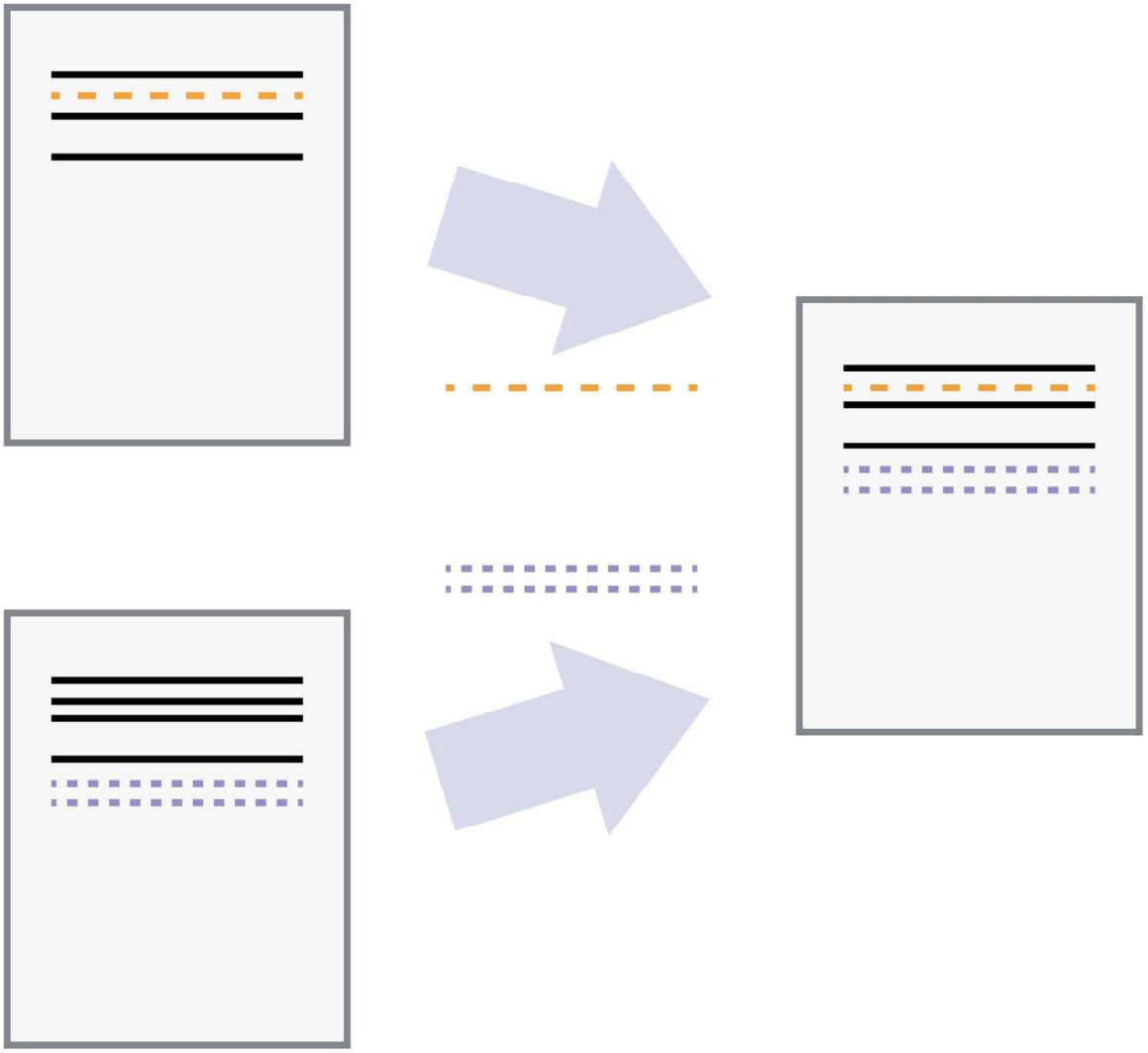


Version Control with Git: Key Concepts

- Git is a toolkit for Automated Version Control
- We need to do some initial work to configure the Git software for each computer.
- Git works at the level of a folder/directory of files.
- We start using Git when we 'initialise a repository' and we 'add' files we want to do version control with.
- From then on, Git tracks changes to those files.
- We control the snapshots / save points / check-points of every file that is being tracked.
- Git allows us to write useful explanations of what we did, so we have a history of our work.
- Git is designed for sharing code over the internet.
- We have tools to help us agree on changes made by more than one person.







Automated Version Control: Key Concepts

- Automated version control is the **lab notebook** of the digital world
- A version control system is a tool that tracks changes for us, like an **unlimited ‘undo’**.
- Version control also allows many people to work in parallel
  - and is equally useful for an individual
- Version control is really a way of recording **metadata** about your work.

Setting Up Git: `git config`

```
$ git config --global user.name "Vlad Dracula"
$ git config --global user.email
“vlad@tran.sylvan.ia"

$ git config --global core.editor
“c:/Windows/System32/notepad.exe"

$ git config --global init.defaultBranch main

$ git config --list
```

git init

- Initialise / create a **repository**.
  - meaning: set up a folder so it is ready for tracking changes using Git.

- A *repository* is a folder where a version control system stores the full history of a project, alongside the project files.
- Git stores all of its repository data in the *.git* directory (or folder).
  - This directory is often hidden in the filesystem.

Staging

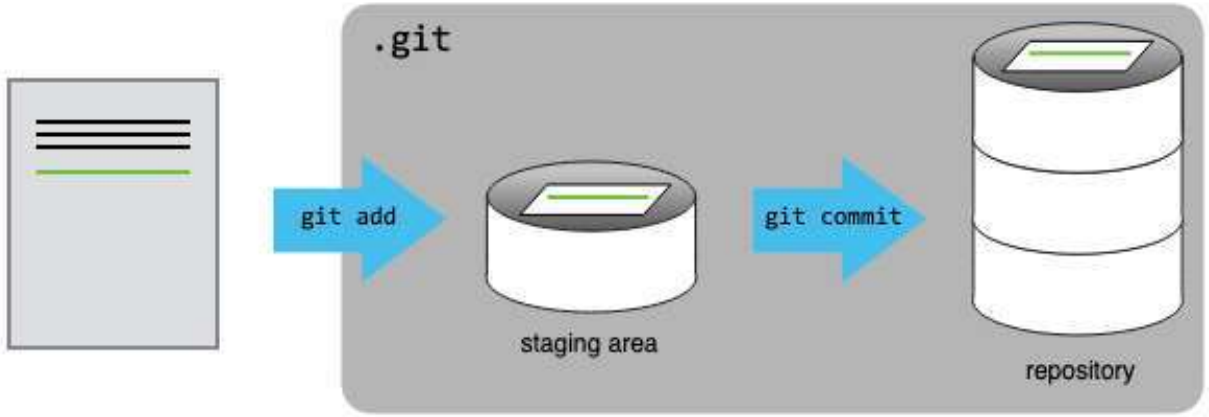
- Suppose we have a file called `mars.txt`
- We will now tell Git that we want to *start tracking changes* for our new file.
- Do this for our new file `mars.txt` by ***add***ing it to our set of *Staged* files.

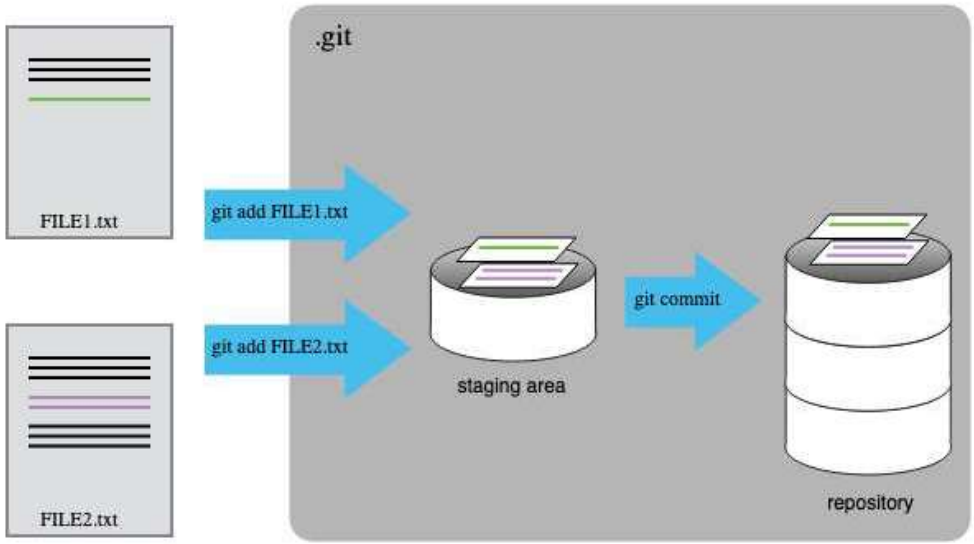
```
git add mars.txt
```

- Git now knows that it's supposed to keep track of `mars.txt`, but it hasn't recorded these changes as a *commit* yet.

Commits

- Before we get Git to record the change, we need to add a description of the changes.
- This is called a *commit message* for adding the file `mars.txt`
  - **git commit –m "Start notes on Mars base."**
- Once done, we can now record the changes along with our message to our future selves by *committing*.
- When we run `git commit`, Git takes everything we have told it to save by using `git add` and stores a copy permanently inside the special *.git* directory.
- This permanent copy is called a *commit* (or *revision*)


.git
git add
staging area
git commit
repository

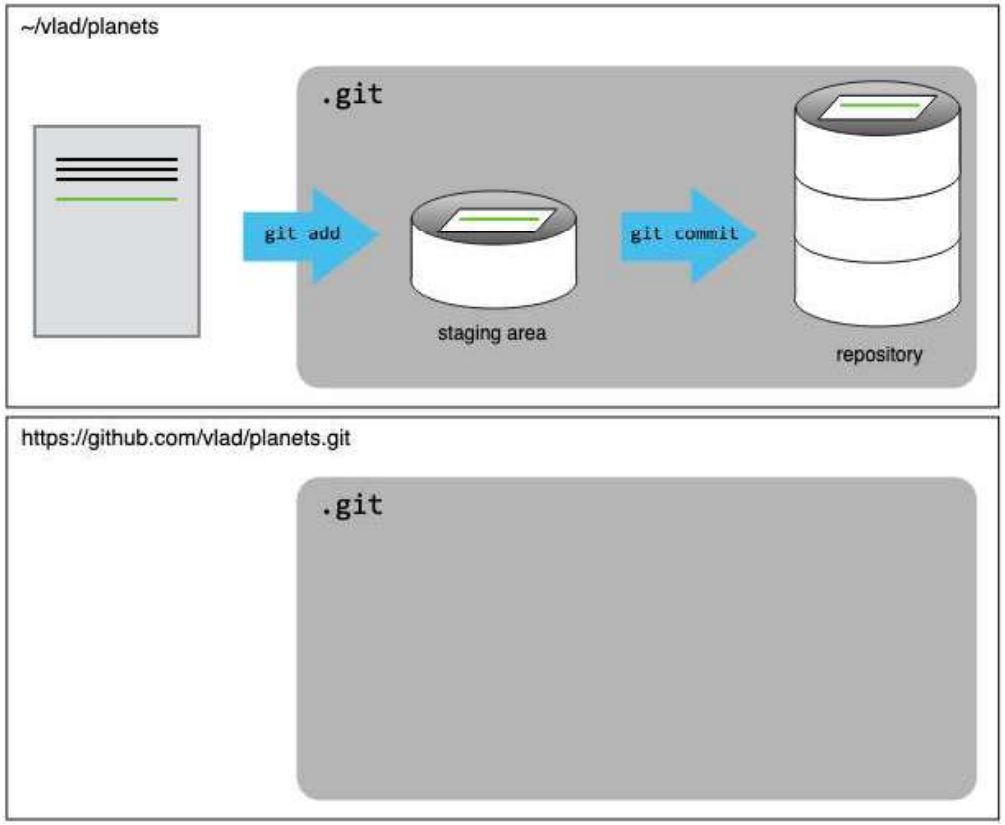

.git
FILE1.txt
FILE2.txt
git add FILE1.txt
git add FILE2.txt
staging area
git commit
repository

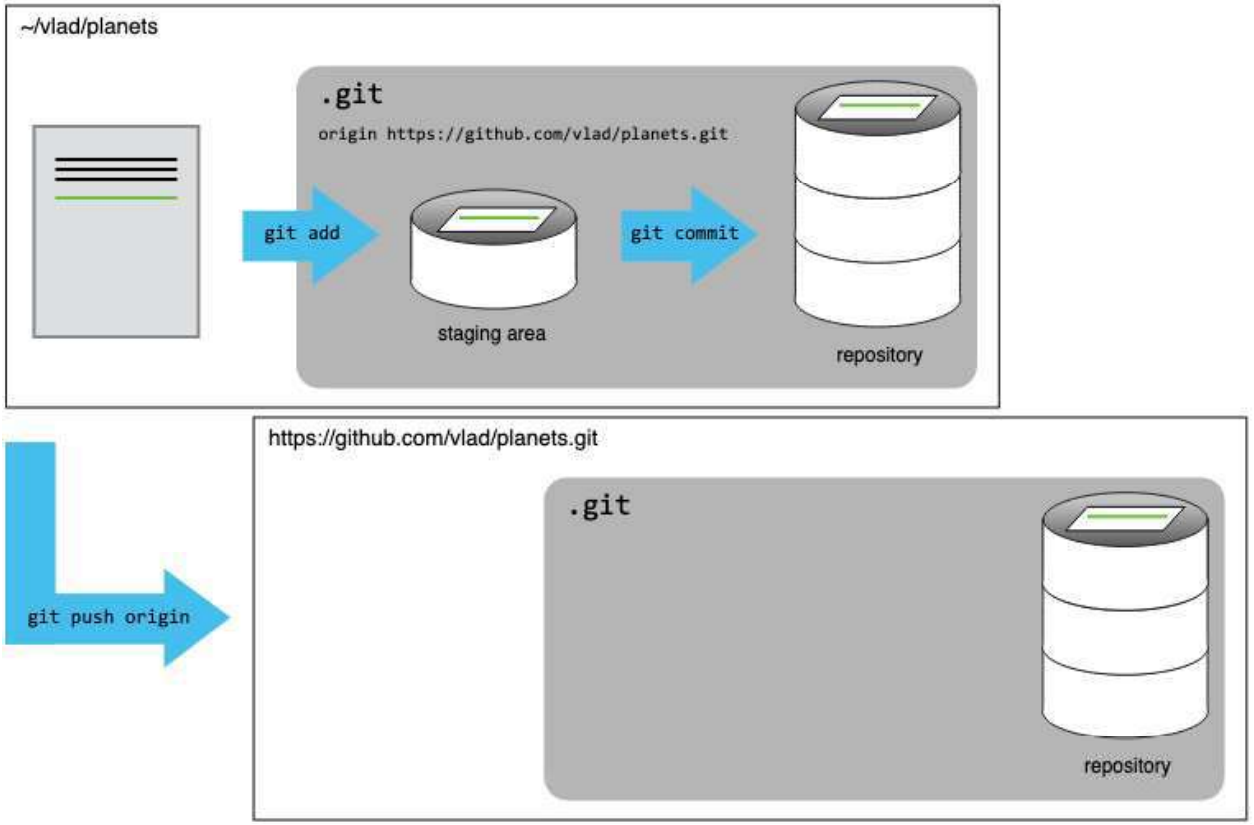
Remotes and cloning

- A local Git repository can be connected to one or more remote repositories.
  - The SSH protocol is used to connect to remote repositories.

- `git push`
  - copies changes from a local repository to a remote repository.

- `git pull`
  - copies changes from a remote repository to a local repository.


~/vlad/planets
.git
git add
staging area
git commit
repository
https://github.com/vlad/planets.git
.git


~/vlad/planets
.git
origin https://github.com/vlad/planets.git
git add
staging area
git commit
repository
git push origin
https://github.com/vlad/planets.git
.git
repository

`push` vs. `commit`

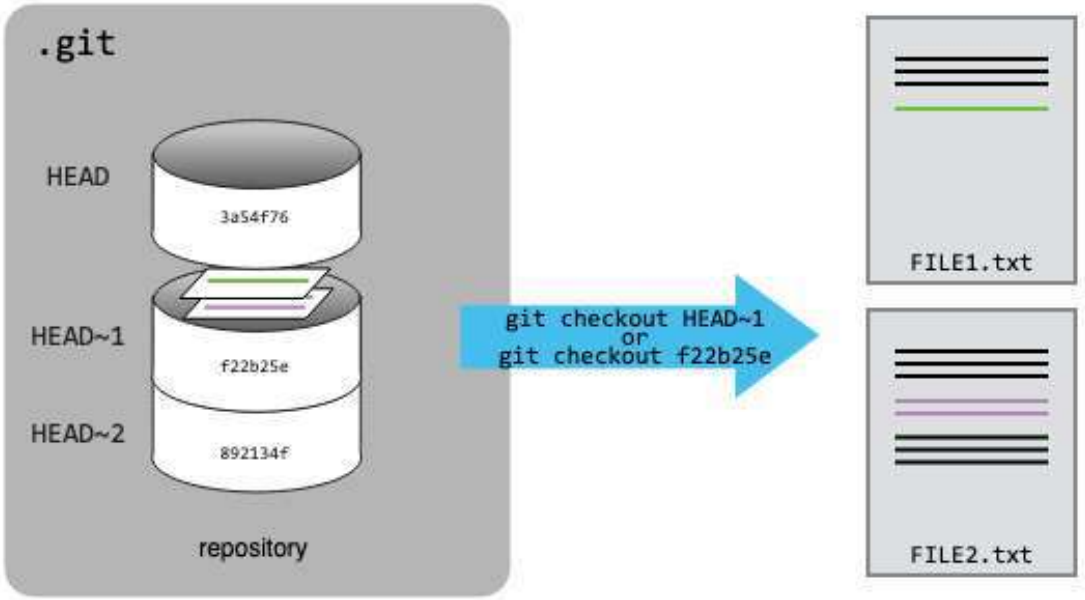
- When we `push` changes, we're interacting with a remote repository to update it with the changes we've made locally

- This corresponds to sharing the changes we've made with others.

- `commit` only updates your local repository with your own changes.

Unlimited undo: `git checkout`

- We can go ‘back in time’ to a previous state of our project.
- This is very useful if
  - we’ve introduced a bug or error
  - want to reproduce our our analysis at a specific point in time.
- We can get the complete state of the project at a specific commit point by using a unique identifier or relative to the current state (or `HEAD`)



Demonstrating an IDE with Git integration

Repo with notebooks

- Install VSCode
  - https://code.visualstudio.com/
- Install Julialang
  - https://julialang.org/
- Download notebooks (check out the whole repo)
  - https://github.com/Lauraz48/EPICS_TUT_2026

Tutorial

- Coding tutorials focus on 2 things
  - Running a power flow for a network directly using OpenDSSDirect.jl
  - PowerModelsDistribution -> set up an optimisation problem for same network
    - Showcase some of the benefits

- Notebooks
  - https://github.com/Lauraz48/E_tut_ongoing

YouTube resources

- JuliaLang
  - https://www.youtube.com/c/TheJuliaLanguage
- PMD
  - https://www.youtube.com/watch?v=LAxytFlle9E
- PMDSE
  - https://www.youtube.com/watch?v=Bl3FsP_6WuM
- InfrastuctureModels
  - https://www.youtube.com/watch?v=POOt1FCA8LI
- JuMP-dev 2025
  - https://www.youtube.com/watch?v=mbIbjj2waxE&list=PLP8iPy9hna6RLvzHeEVCwkUkZV3aChElQ

OpenDSS video training

- https://www.epri.com/events/d2a46142-33e3-4cf5-9d12-ef8d31c9f84f -> attachments -> media
- https://www.epri.com/events/c97292af-0790-4514-b9df-ccef80d685e1 -> attachments -> media
- Prof. Nando Ochoa has various recordings linked here 🙏 https://sites.google.com/view/luisfochoa/invited-talks-webinars-and-tutorials

Distribution network modeling text books

- T.A. Short, Electric Power Distribution Handbook, Second Edition, CRC Press, ISBN9781466598652
- James J. Burke, Power Distribution Engineering: Fundamentals and Applications, CRC Press, ISBN 9780824792374
- Turan Gonen, Electric Power Distribution Engineering, Third Edition, CRC Press, ISBN 9781482207002
- J. Duncan Glover, Thomas Overbye, Mulukutla S. Sarma, Power System Analysis and Design, Sixth Edition, Cengage, ISBN 9781305636200
- Luces M. Faulkenberry, Walter Coffer, Electrical Power Distribution and Transmission, Prentice Hall, ISBN 9780132499477
- James Momoh, *Smart Grid: Fundamentals of Design and Analysis, 1st Edition*, Wiley, ISBN 978-0470889398
- Steven Low's lecture notes, https://netlab.caltech.edu/book/

PMD related publications

- Geth, F., Heidari, R., Chapman, A. & Clark, J., Considerations and Design Goals for Unbalanced Optimal Power Flow Benchmarks, PSCC 2024
- Heidari, R., and Geth, F., Improved Algebraic Inverter Modelling for Four-Wire Power Flow Optimization, PSCC 2024
- Vanin M., Geth F., D'hulst R, & Van Hertem, D. "Combined Unbalanced Distribution System State and Line Impedance Matrix Estimation", Int. J. Electrical Power & Energy Systems
- Geth F., Claeys S., & Heidari R. "On the Implementation of the Fixed Point Iteration Current Injection Method to Solve Four-Wire Unbalanced Power Flow in PowerModelsDistribution.jl"
- Geth F. "Pitfalls of Zero Voltage Values in Optimal Power Flow Problems", IEEE PES General Meeting 2023
- Geth F., Heidari R. & Koirala A. "Computational analysis of impedance transformations for four-wire power networks with sparse neutral grounding", ACM E-energy 2022
- Geth F. & Liu B., "Notes on BIM and BFM optimal power flow with parallel lines and total current limits", IEEE PES General Meeting 2022
- Claeys, S., Geth, F., Sankur, M., & Deconinck, G. (2021). No-load linearization of the lifted multi-phase branch flow model: equivalence and case studies. IEEE ISGT Europe, October.
- Van Acker, T., Geth, F., Koirala, A., & Ergun, H., General **polynomial chaos** in the current-voltage formulation of the optimal power flow problem, EPSR 2022
- Geth, F. & Van Acker, T., **Harmonic** optimal power flow with **transformer excitation**, EPSR 2022
- Claeys, S., Geth, F., & Deconinck, G. Optimal power flow in **four-wire** distribution networks: formulation and benchmarking. EPSR 2022
- Claeys, S., Vanin, M., Geth, F., & Deconinck, G., "Applications of optimization models for electricity distribution networks", Wiley Interdisciplinary Reviews: Energy and Environment, vol. 10, no. 5, pp. e401, September/October 2021
- Claeys, S., Geth, F., & Deconinck, G. (2021). **Line parameter estimation** in multi-phase distribution grids without voltage angle measurements. CIRED Open Access Journal, September
- Geth, F., & Coffrin, C. (2019). "Direct method to **recover current and voltage** in multi-conductor optimal power flow models." In IEEE PES General Meeting Atlanta 2019, Georgia.
- Claeys, S., Geth, F., & Deconinck, G. "Optimization of **gang-operated on-load tap changers** in multi-conductor radial networks: formulation and **convex relaxation**. In CIGRE Chengdu 2019 Symposium (pp. 1–10).
- Fobes, D. M., Coffrin, C., Geth, F., & Claeys, S. (2020). **PowerModelsDistribution.jl**: an open-source framework for exploring distribution power flow formulations. Electric Power Systems Research, 189(December), 106664.
- Geth, F., Claeys, S., & Deconinck, G. (2020). **Nonconvex lifted unbalanced branch flow model**: derivation, implementation and experiments. Electric Power Systems Research, 189(December), 106558.
- Geth, F., Coffrin, C., & Fobes, D. M. (2020). A flexible **storage** model for power network optimization. ACM E-Energy, 1–6. http://arxiv.org/abs/2004.14768
- Claeys, S., Deconinck, G., & Geth, F. (2021). **Voltage-dependent load models i**n unbalanced optimal power flow using power cones. IEEE Trans. Smart Grid (Prepublished).
- Claeys, S., Vanin, M., Geth, F., & Deconinck, G. (2021). **Applications of optimization models** for electricity distribution networks. Wiley Interdisciplinary Reviews: Energy and Environment, e401.

References on software

- Gao, D. W., Muljadi, E., Tian, T., & Miller, M. (2017). **Software comparison** for renewable energy deployment in a distribution network. Golden, CO. Retrieved from https://www.nrel.gov/docs/fy17osti/64228.pdf
- Meinecke, S., Braun, M., Meier, F., Scheidler, A., Schafer, F., Dollichon, J., … Menke, J.-H. (2018). **Pandapower -** An open-source python tool for convenient modeling, analysis, and optimization of electric power systems. IEEE Trans. Power Syst., 33(6), 6510–6521. https://doi.org/10.1109/tpwrs.2018.2829021
- Rigoni, V., & Keane, A. (2020). **Open-DSOPF** : an open-source optimal power flow formulation integrated with OpenDSS,
- Dugan, R. C., & McDermott, T. E. (2011). An open source platform for collaborating on smart grid research (**OpenDSS**). IEEE Power Energy Soc. General Meeting, 1–7. https://doi.org/10.1109/PES.2011.6039829
- Morstyn, T., Collett, K. A., Vijay, A., Deakin, M., Wheeler, S., Bhagavathy, S. M., … Mcculloch, M. D. (2020). **OPEN** : An open-source platform for developing smart local energy system applications. Applied Energy, 275(June), 115397. https://doi.org/10.1016/j.apenergy.2020.115397
- Fobes, D. M., Coffrin, C., Geth, F., & Claeys, S. (2020). **PowerModelsDistribution**. jl: an open-source framework for exploring distribution power flow formulations. Electric Power Systems Research, 189(December), 106664.
- Coffrin, C., Bent, R., Sundar, K., Ng, Y., & Lubin, M. (2018). **PowerModels.jl**: an open-source framework for exploring power flow formulations. In Power Syst. Comp. Conf. (Vol. 20, p. 8). Dublin, Ireland.
- Ergun, H., Dave, J., Van Hertem, D., & Geth, F. (2019). Optimal power flow for AC/DC Grids: formulation, convex relaxation, linear approximation and implementation (**PowerModelsACDC.jl**). IEEE Trans. Power Syst., 34(4), 2980–2990. https://doi.org/10.1109/tpwrs.2019.2897835
- Claeys, S., Vanin, M., Geth, F., & Deconinck, G. (2020). Applications of optimization models for electricity distribution networks. Submitted to WIREs Energy and Environment.
- Czekster, R. M. (2020). Tools for modelling and simulating the Smart Grid. ArXiv [Cs.PF].
- Henriquez-Auba, R., Lara, J. D., Roberts, C., Pallo, N., & Callaway, D. S. (2020). **LITS.jl** — An open-source Julia based simulation toolbox for low-inertia power systems. ArXiv, 1–8.
- Plietzsch, A., Kogler, R., Auer, S., Merino, J., Gil-de-Muro, A., Liße, J., … Hellmann, F. (2021). **PowerDynamics.jl** -- An experimentally validated open-source package for the dynamical analysis of power grids. Retrieved from http://arxiv.org/abs/2101.02103
- Vanin, M., Van Acker, T., D'hulst, R., & Van Hertem, D. (2020). A framework for constrained static state estimation in unbalanced distribution networks, (**PowerModelsDistributionStateEstimation.jl**) 1–8. Retrieved from http://arxiv.org/abs/2011.11614