

How to Build a Synapse with MCell/CellBlender: A Computational Microscope for Spatially Realistic Simulation of Cellular Micophysiology

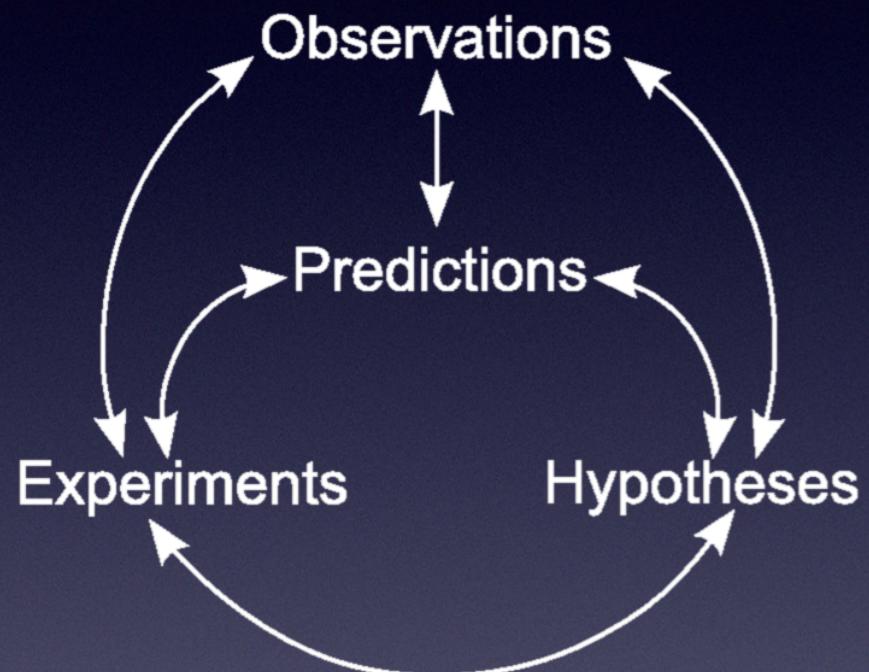
Thomas M. Bartol

Computational Neurobiology Laboratory
The Salk Institute

bartol@salk.edu
mcell.org

Scientific Computing Enables The Scientific Discovery Cycle

Scientific Discovery Cycle



Types of Experiments & Models

Thought Experiments Mental Models	Bench Experiments Physical World Model Systems Model Organisms	Computational Experiments Computational Models
--------------------------------------	---	---

“What I cannot create, I do not understand” -- *Richard Feynman*

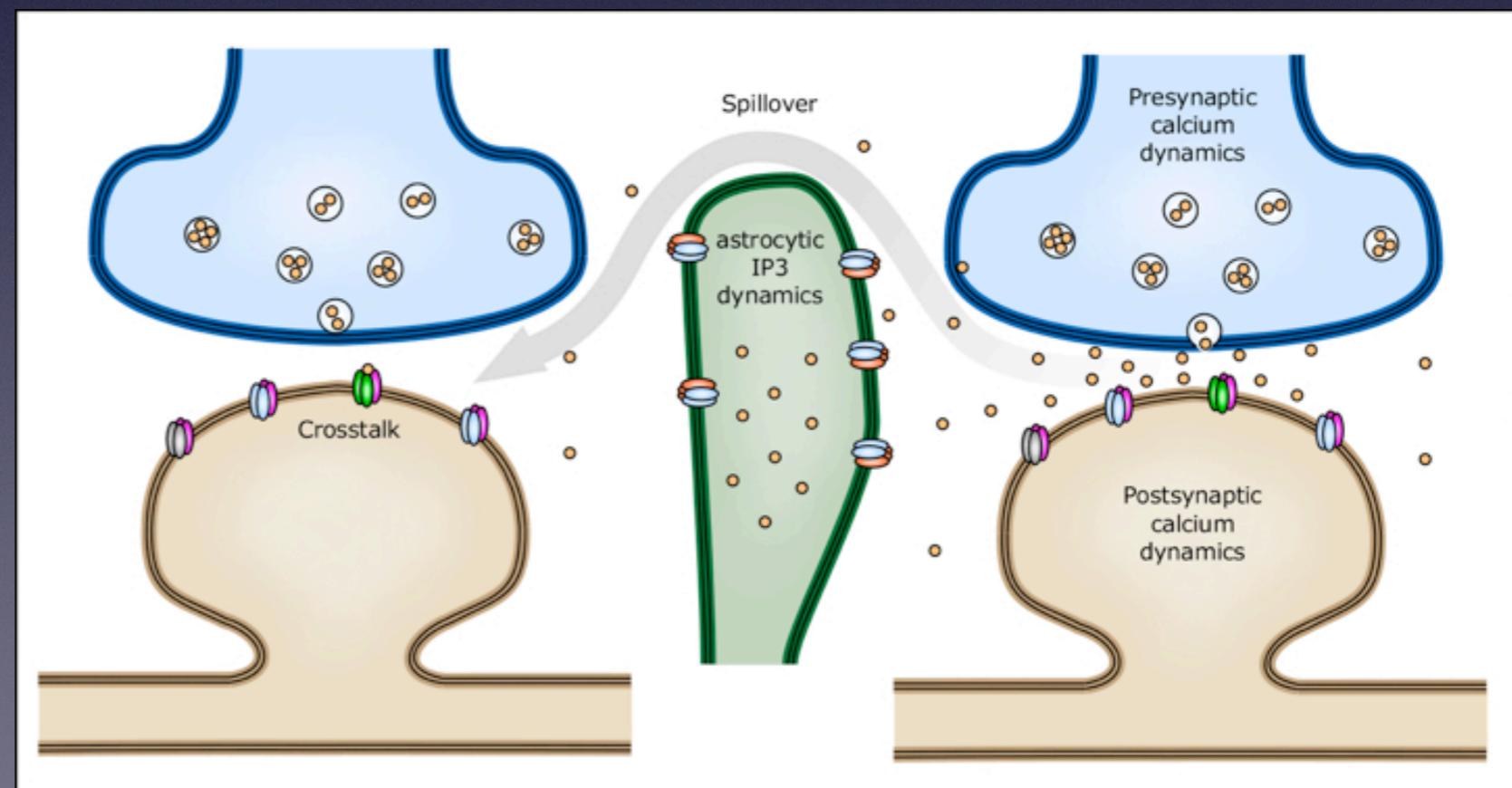
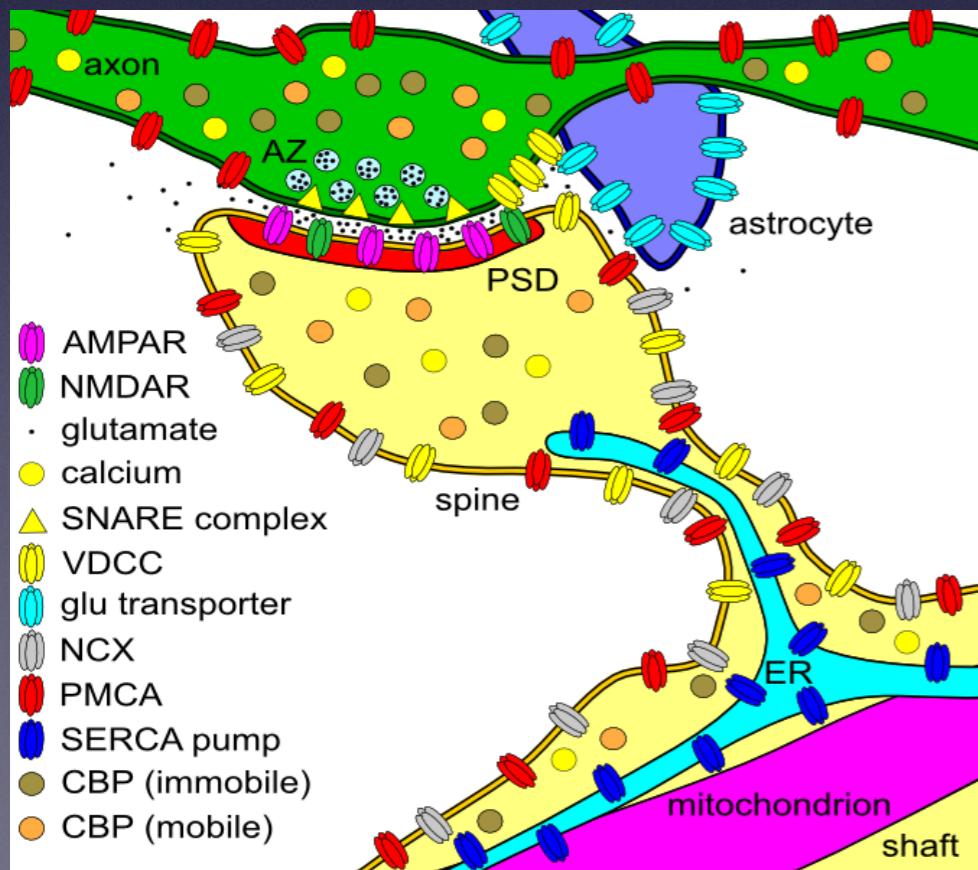
Philosophical Approach: Understanding through simulation

Motivation

How do synapses work?

How does synaptic structure influence synaptic function?

- Topology of synaptic and perisynaptic space (neuropil !!!)
- Organization of pre- and postsynaptic cytoplasmic organelles
- Distributions of receptors, ion channels, enzymes, transporters...
- Biochemical reaction networks and their dynamics
- Cells are not well mixed & numbers of molecules can be small
- Simulations must include stochastic spatiotemporal dynamics



MCell: Monte Carlo Simulator of Cellular Micophysiology

- Models realistic 3D reaction/diffusion systems
- Rigorously validated and highly optimized stochastic Monte Carlo methods
- Tracks Brownian dynamics diffusion and reaction of individual particles in 3D volumes and on 2D surfaces embedded in 3D
- Arbitrarily complex 3D geometry -- triangle surface meshes
- Arbitrarily complex reaction networks -- Markov processes and Network-free rule-based specification modeled as discrete event-driven point processes
- Flexible and Powerful Python API
- Sophisticated model building, visualization and analysis environment -- CellBlender

Diffusion: Brownian Motion

Thermal Velocity:

$$\bar{v} \approx \sqrt{\frac{3kT}{m}}$$

For Water:

$$\bar{v} \approx \sqrt{\frac{3(1.3807 \times 10^{-23})(298)}{3 \times 10^{-26}}} \approx 640 \text{ ms}^{-1}$$

Diffusion: Brownian Motion

But, how far do the water molecules go between collisions?

1. Density of water $\approx 1\text{g/cc}$ or 1000g/liter .
2. Molecular weight of water $\approx 18\text{g/mole}$.
3. From 1 and 2, the molarity of water $= 1000/18 = 55.56 \text{ moles/liter}$
4. $55.56 \times 6.022 \times 10^{23} \text{ molecules/mole} =$
 $3.3458 \times 10^{25} \text{ or } 3.3458 \times 10^{22} \text{ molecules/cc}$
5. Average volume of each molecule is:
 $1/3.3458 \times 10^{22} \approx 3.0 \times 10^{-23} \text{ cc/molecule}$
6. Assuming that each molecule corresponds to a spherical space:

$$\frac{\frac{4}{3}\pi r^3}{3} = 3.0 \times 10^{-23} \quad \text{in } 0.3 \text{ ps !!!}$$
$$r = 0.2 \text{ nm}$$

Diffusion: Fick's Second Law

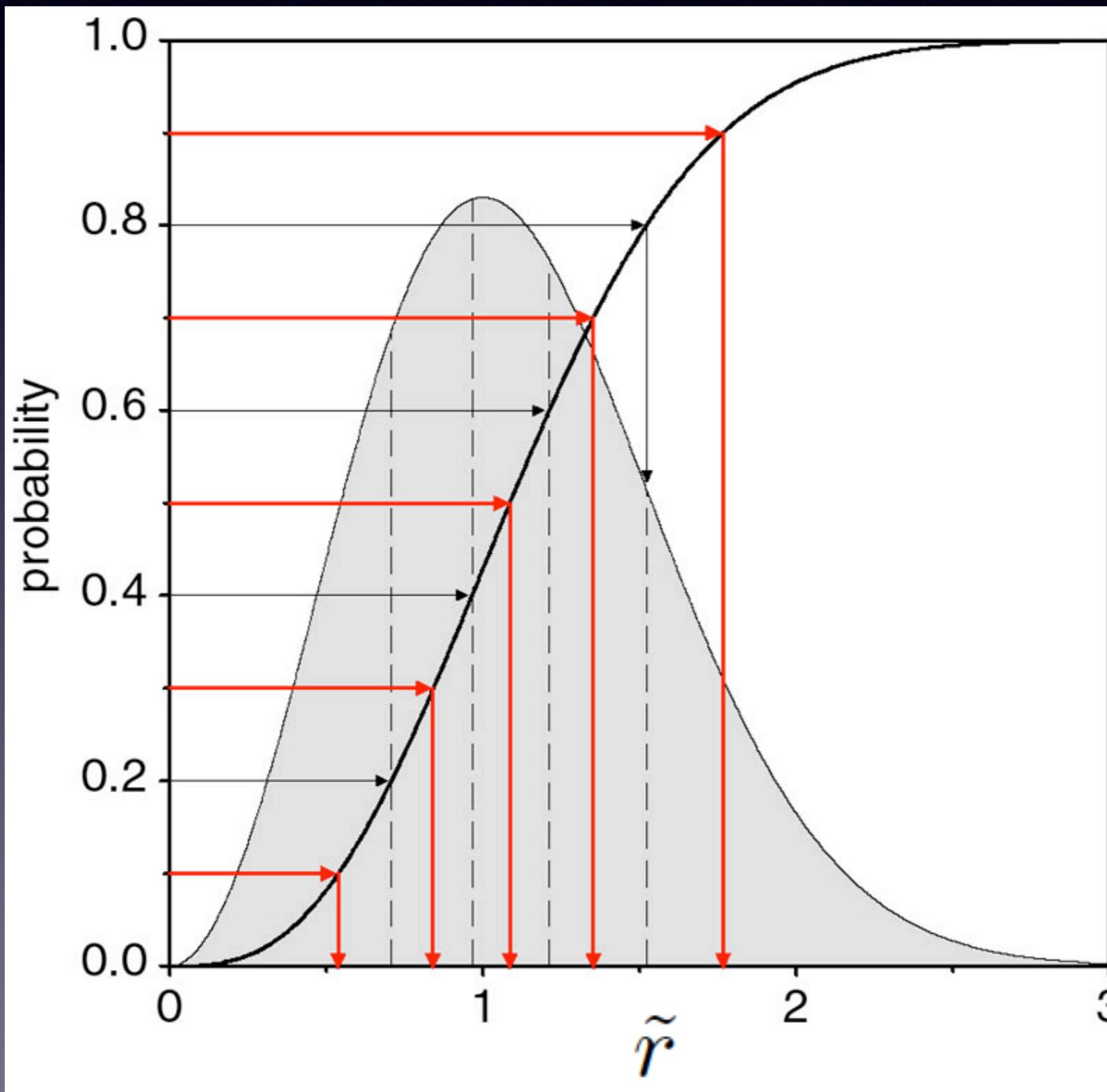
Applies when concentration in volume is changing in time (i.e. $J_{\text{in}} \neq J_{\text{out}}$)

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2}$$

$$\frac{\partial c}{\partial t} = D \nabla^2 c$$

Free Diffusion: Concentration in Space and Time

$$\rho(r, t) = \frac{1}{\pi^{d/2} \lambda^d} e^{-r^2/\lambda^2}$$



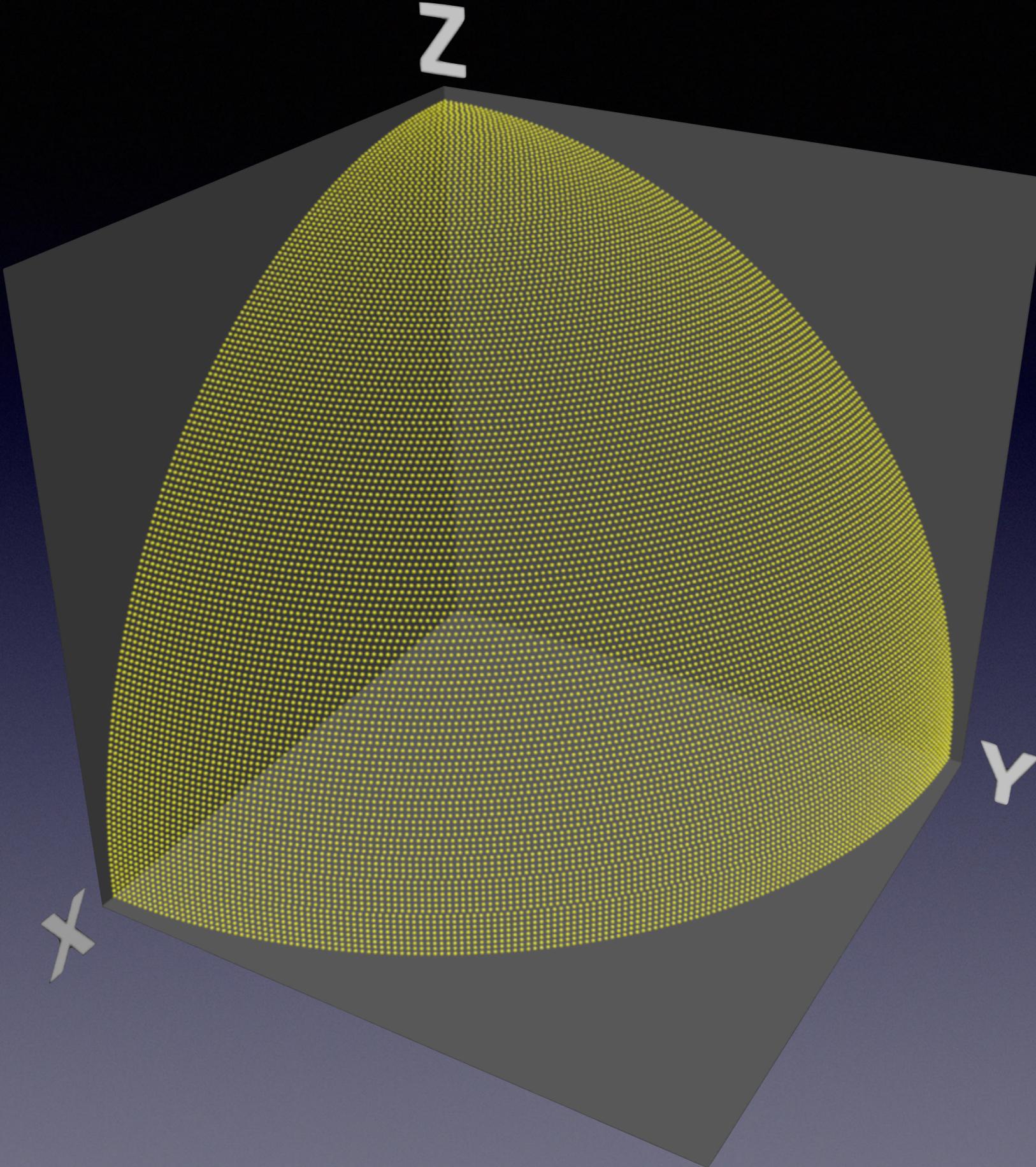
$$\lambda = \sqrt{4Dt}$$

$$\tilde{r} = r/\lambda$$

$$\bar{l}_r = 2 \sqrt{\frac{4D\Delta t}{\pi}}$$

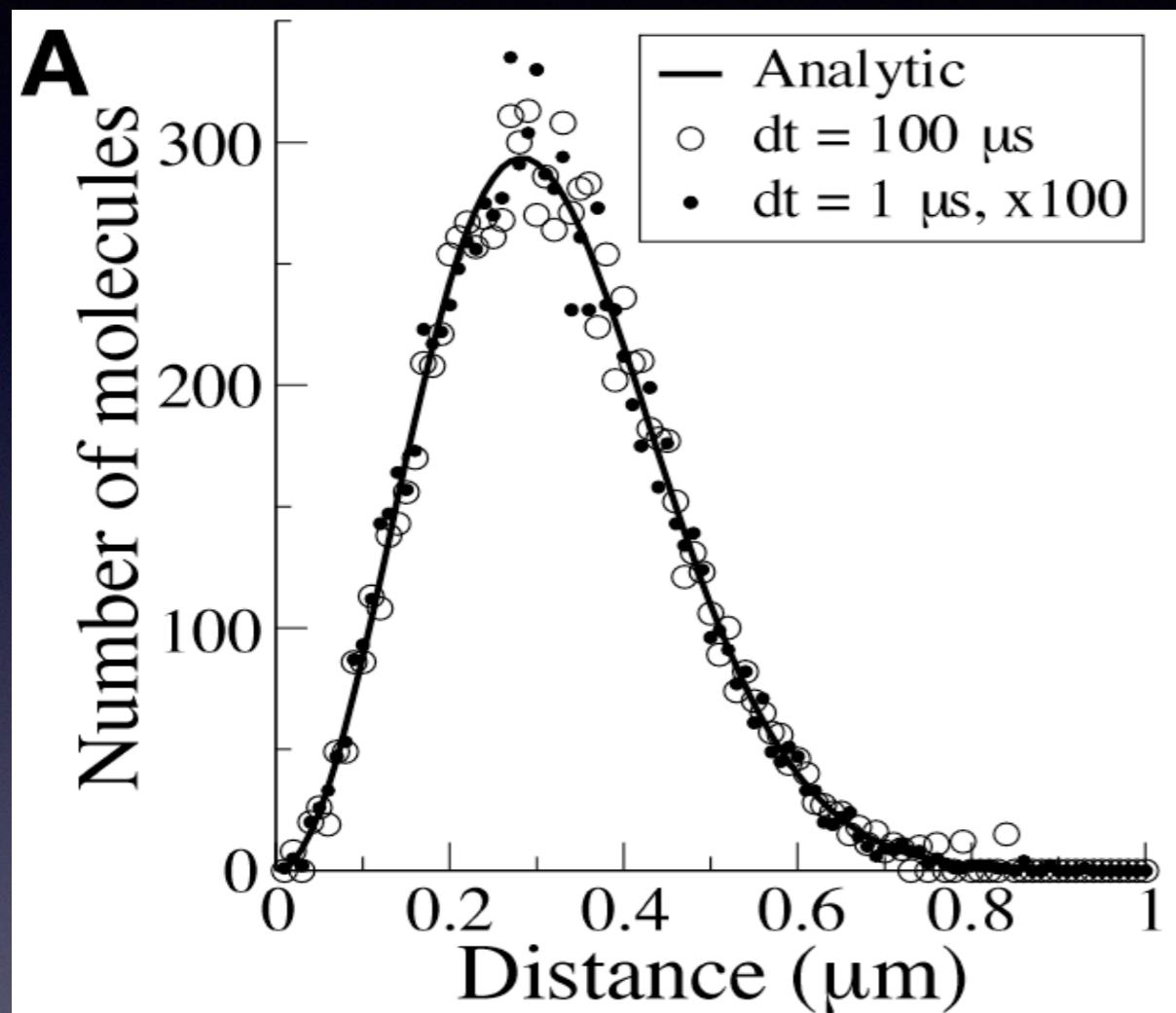
$$\bar{l}_\perp = \sqrt{\frac{4D\Delta t}{\pi}}$$

Diffusion: Random Directions

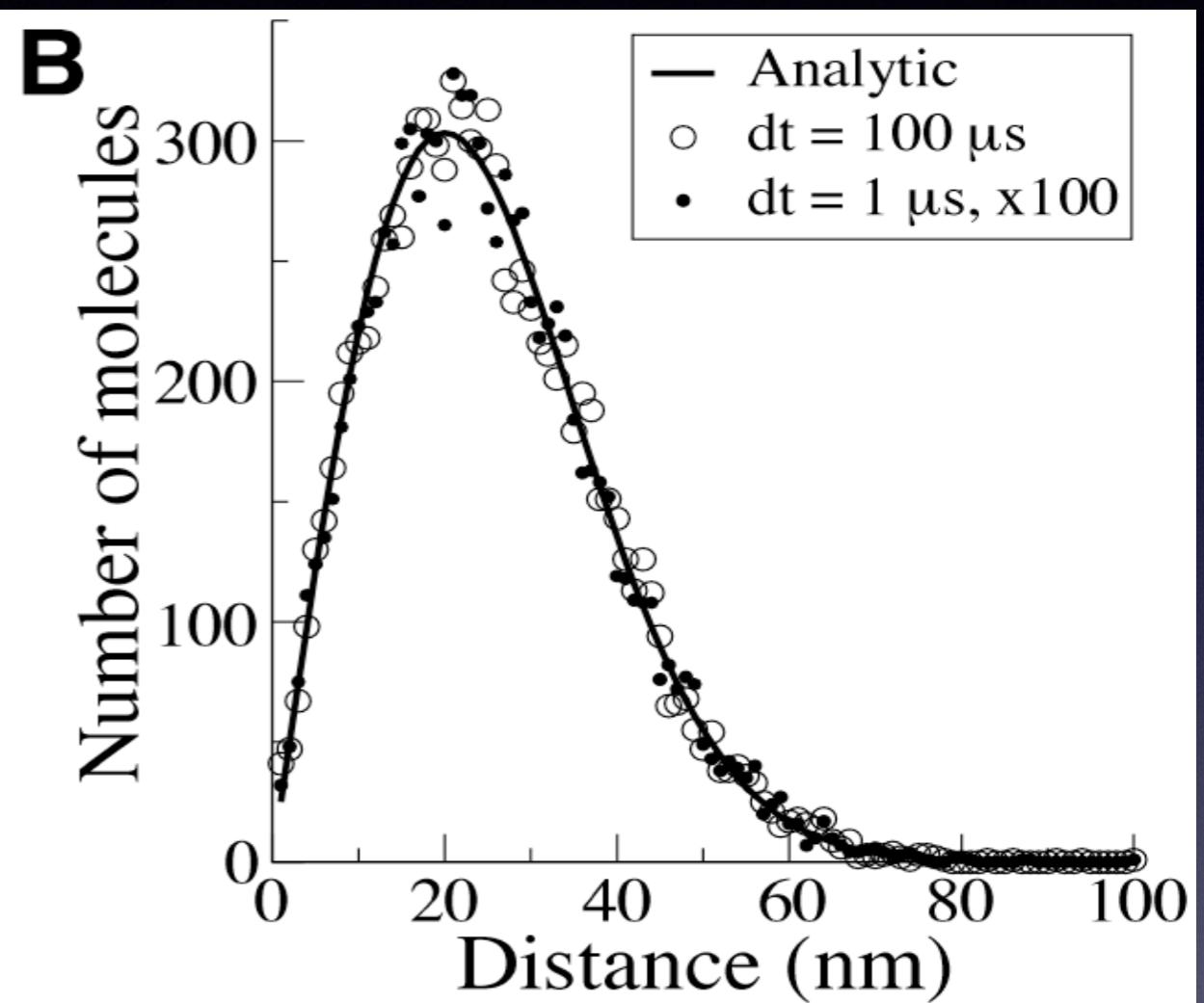


Grid-Free Random Walk Diffusion

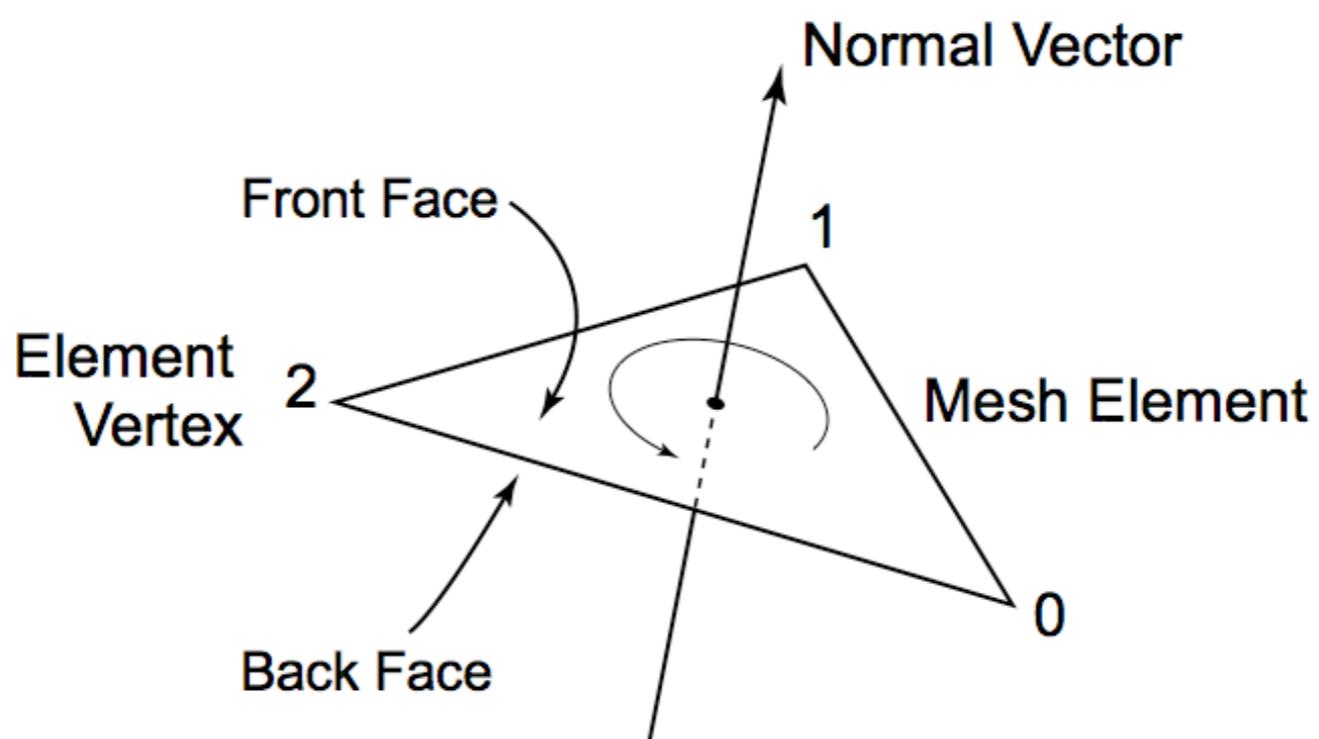
3D diffusion



2D diffusion

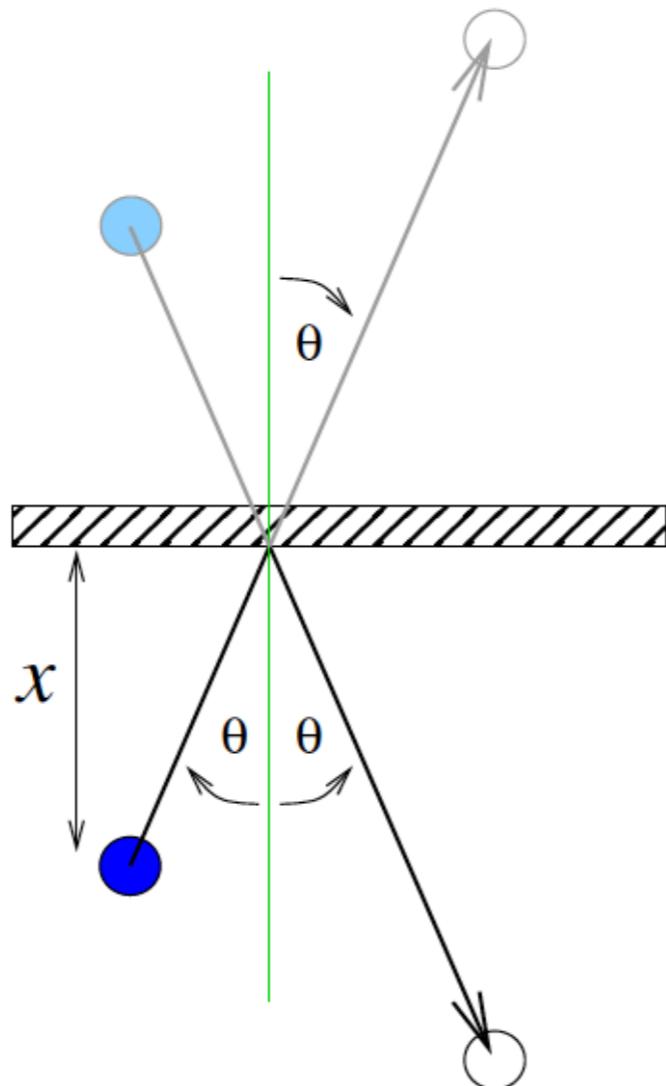


Surface Mesh Representation



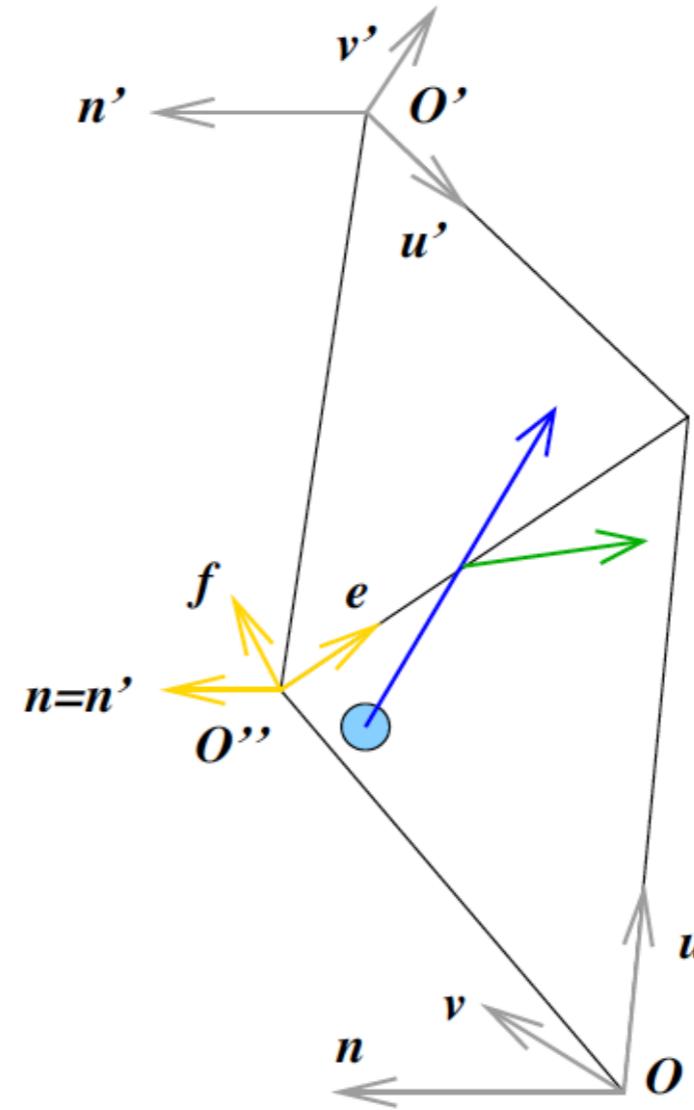
Diffusion with Boundaries

A



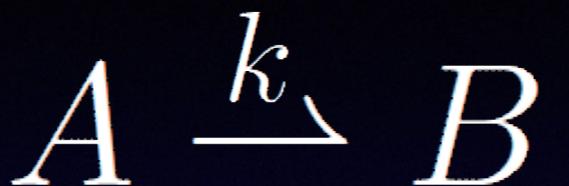
bounded volume diffusion

B



bounded surface diffusion

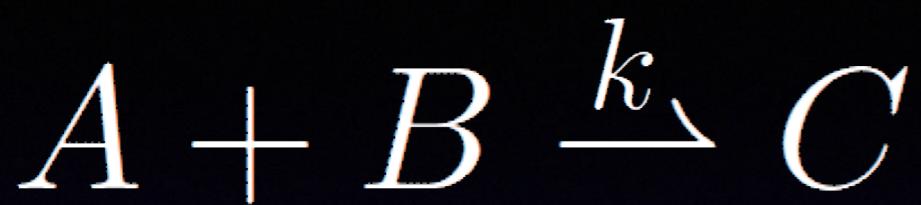
Unimolecular Reactions



Distribution of first-order decay lifetimes is:

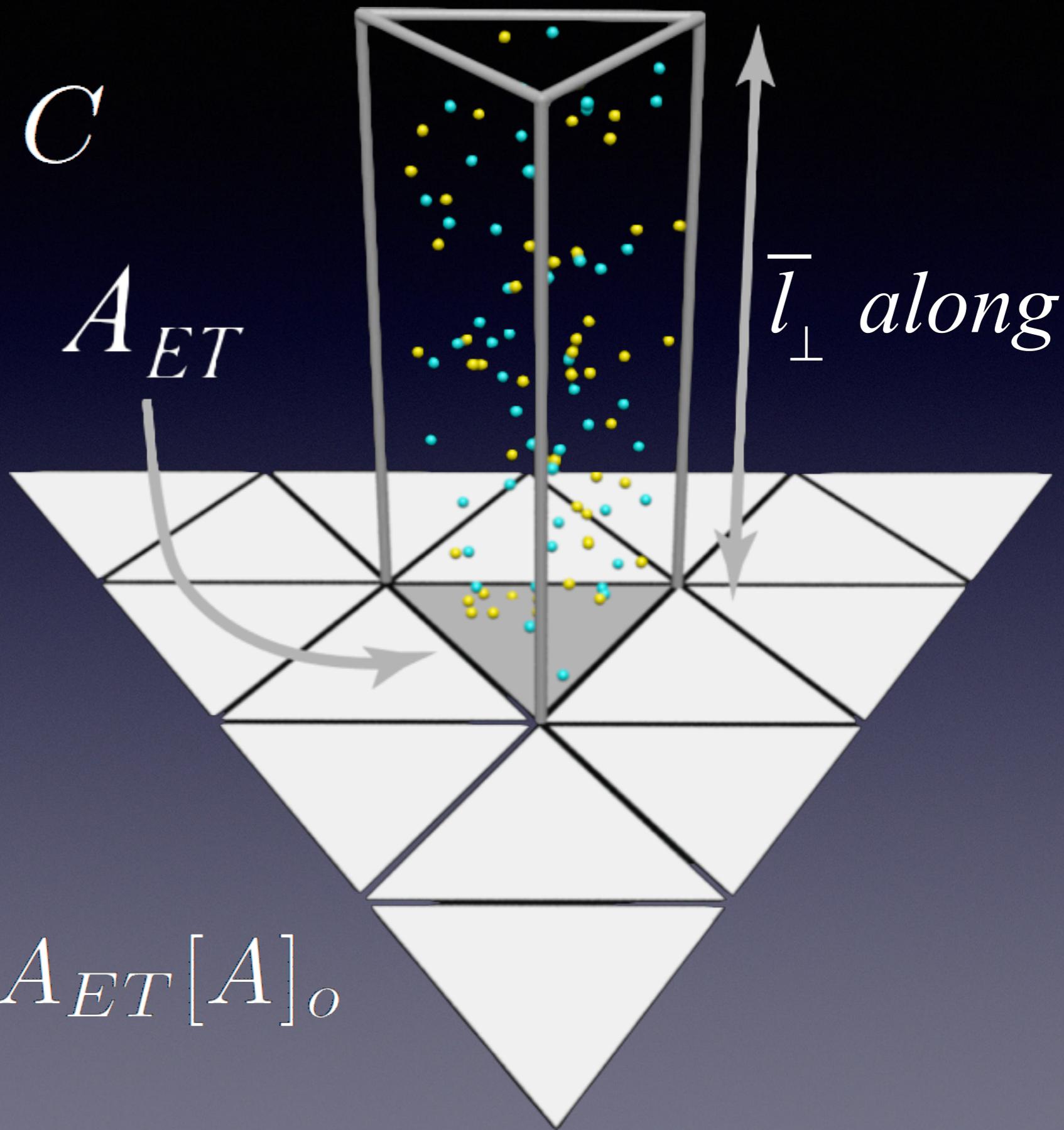
$$\rho(t) = ke^{-kt} \quad t_l = \frac{-\ln(R)}{k}$$

Bimolecular Reactions: Rate of Encounter



$$A_{ET}$$

$$\bar{l}_\perp \text{ along } \hat{n}$$



$$N_H = N_A \bar{l}_\perp A_{ET} [A]_o$$

Bimolecular Reactions: Volume/Surface



Probability of reaction between diffusing volume molecules and a single surface molecule:

From rate of encounter:

$$p_{bt} = 1 - (1 - p_b)^{N_H} \approx N_H p_b$$

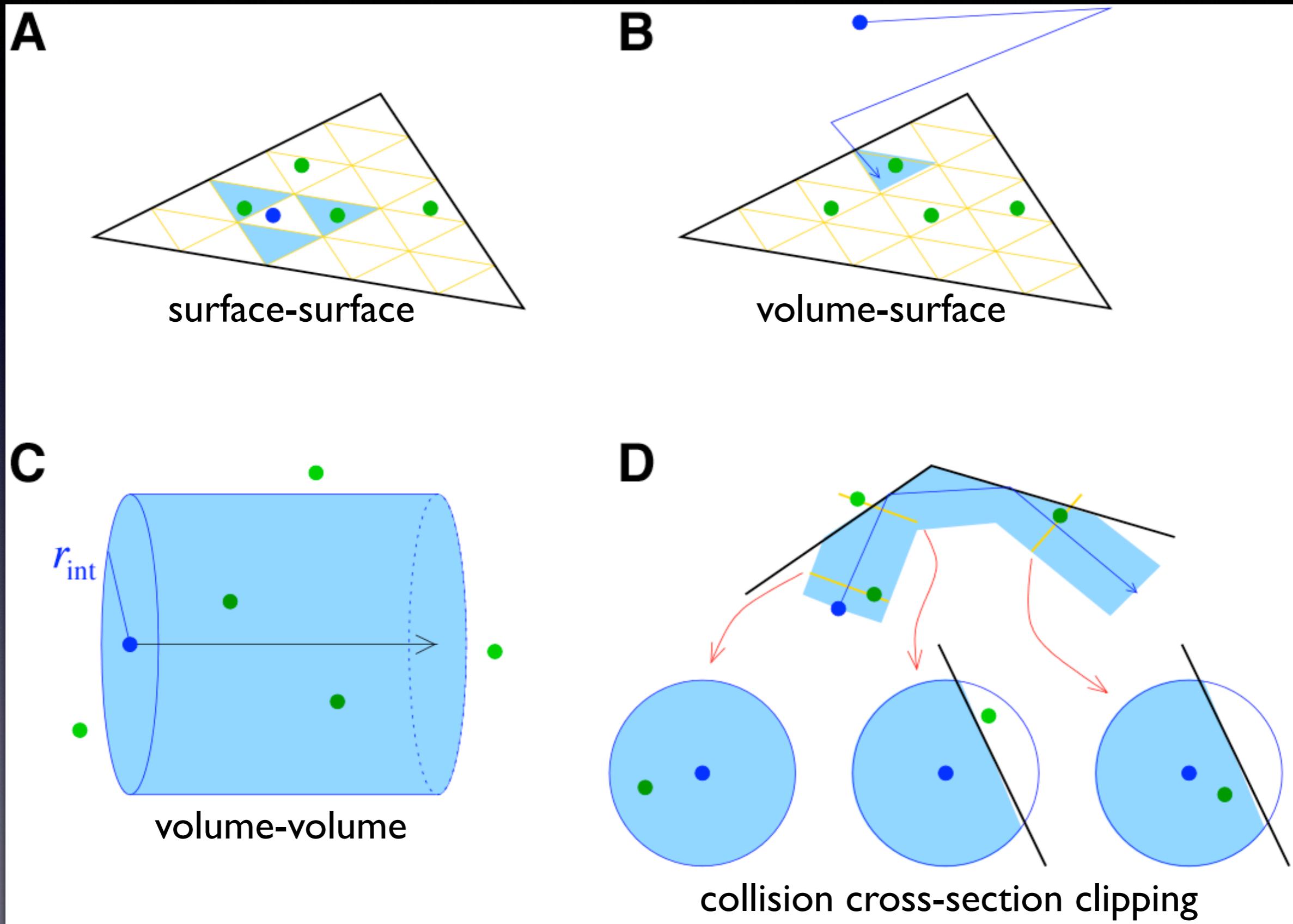
From Mass Action:

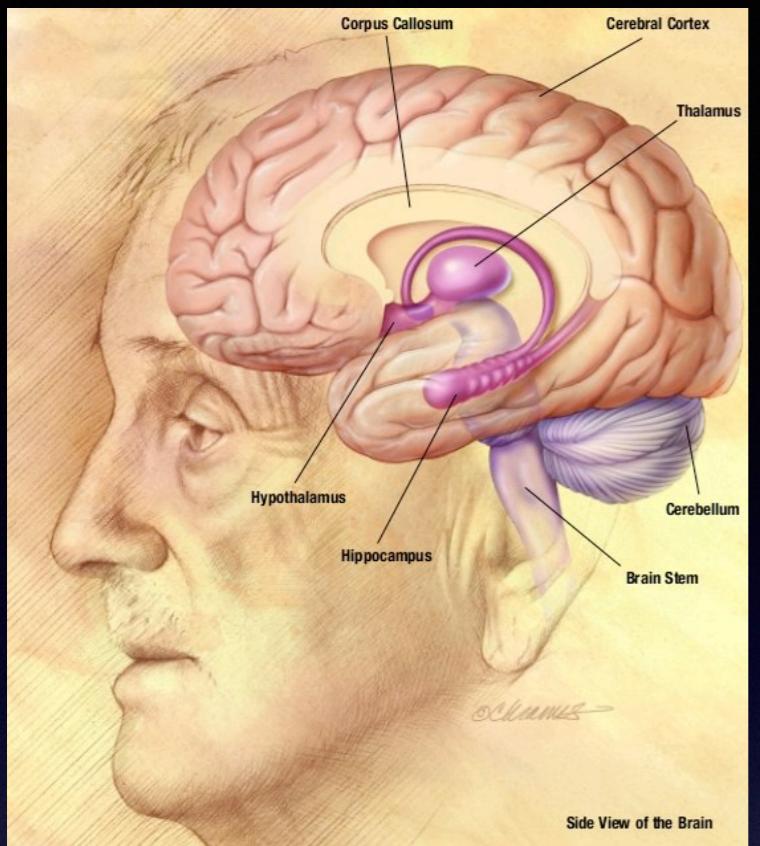
$$p_{bt} = k[A]_o \Delta t$$

$$N_H = N_A \bar{l}_\perp A_{ET} [A]_o$$

$$p_b = \frac{k}{N_A A_{ET}} \sqrt{\frac{\pi \Delta t}{D}}$$

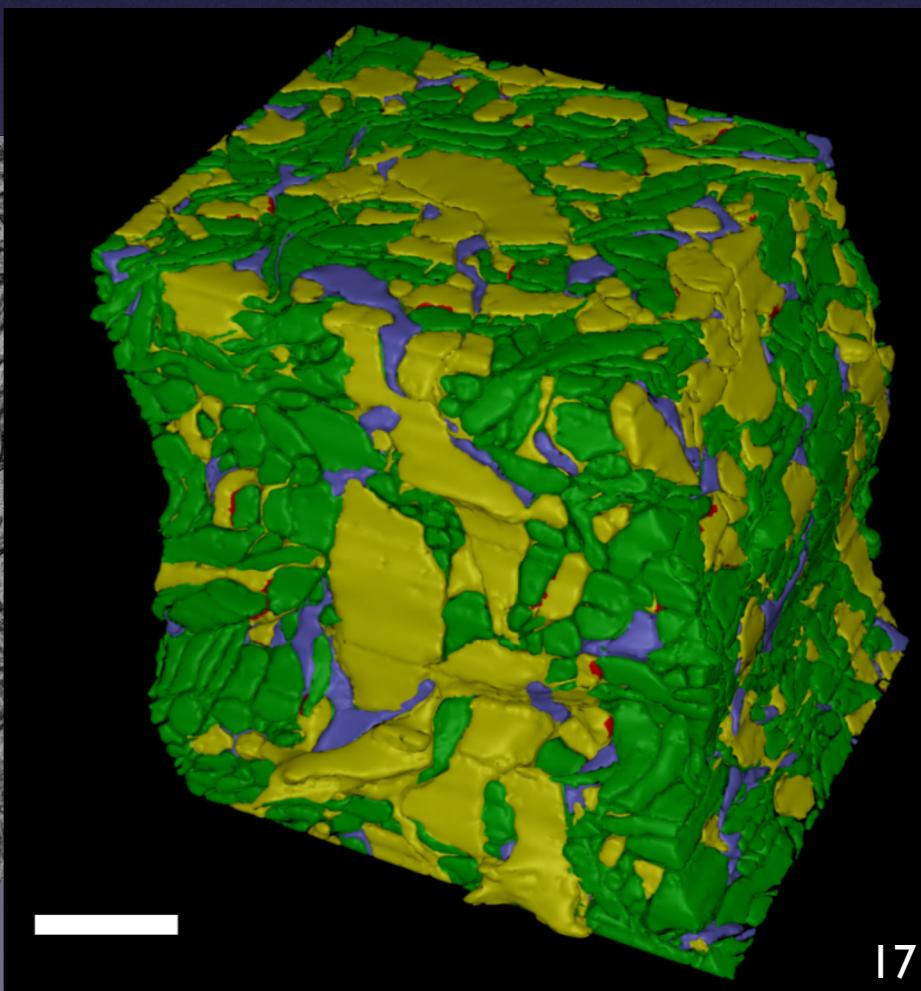
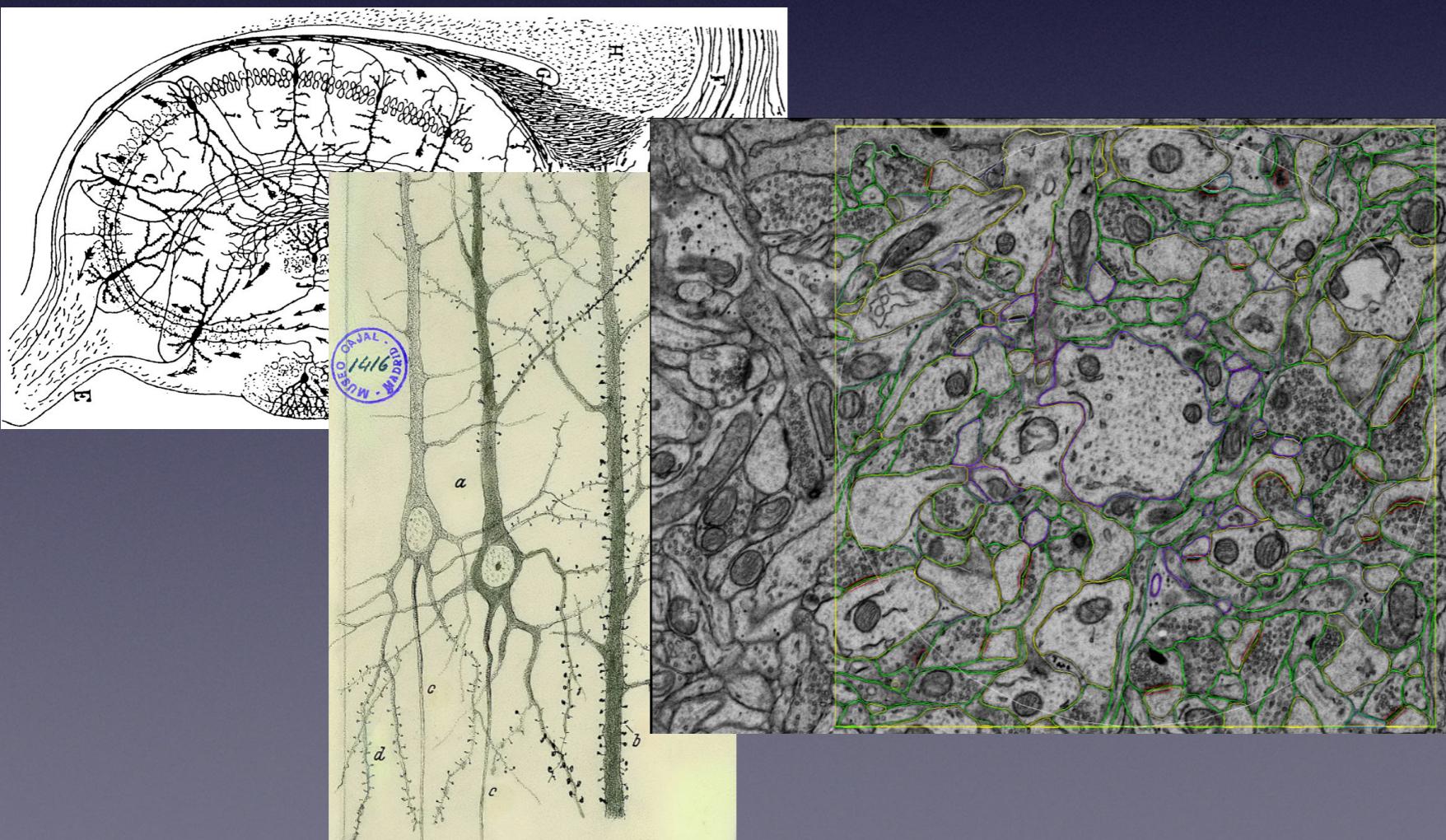
Bimolecular Reactions: Collision Detection





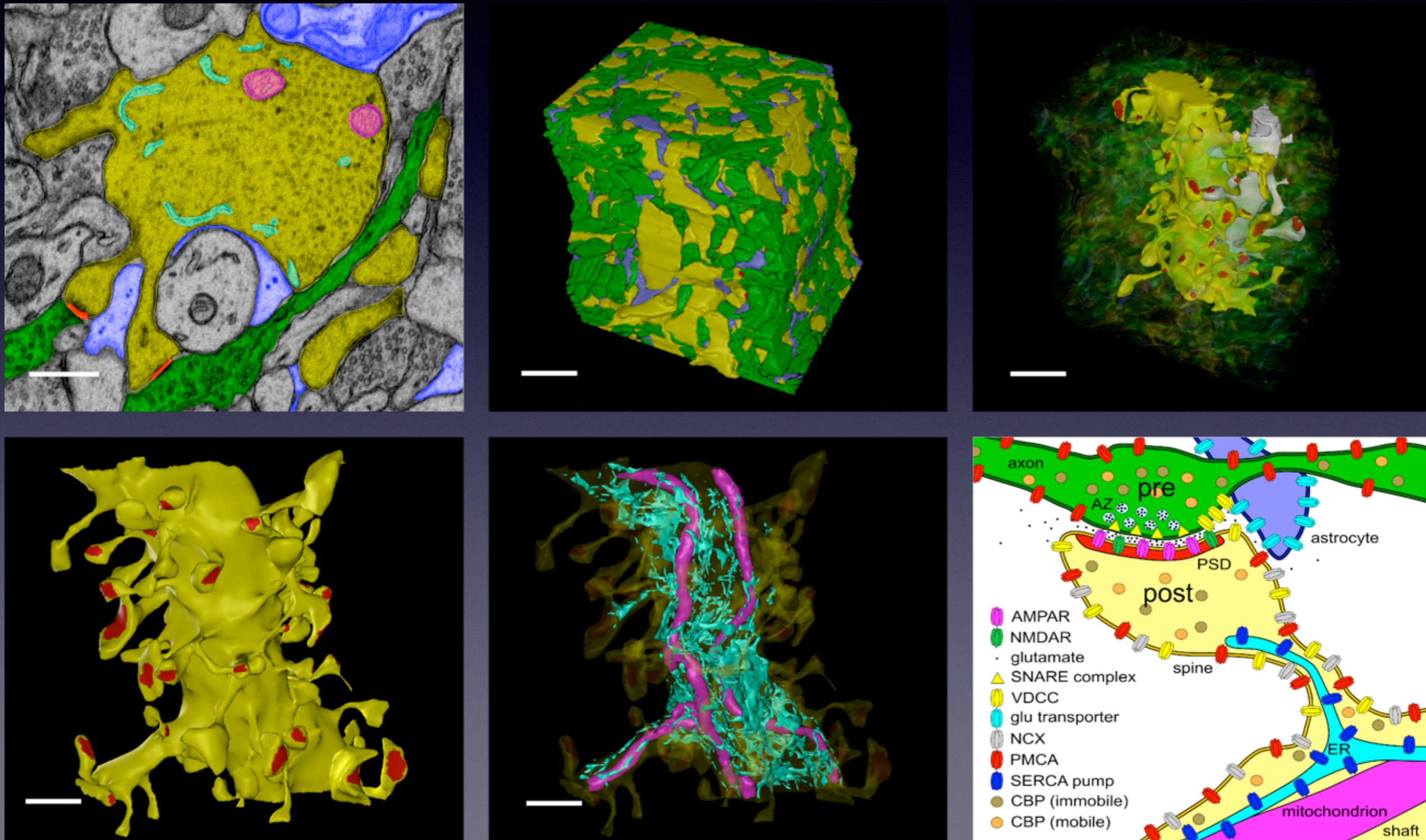
Model Constraints

Realistic synaptic morphology from ssTEM reconstruction of hippocampal neuropil
(collaboration with Kristen Harris, UT Austin)



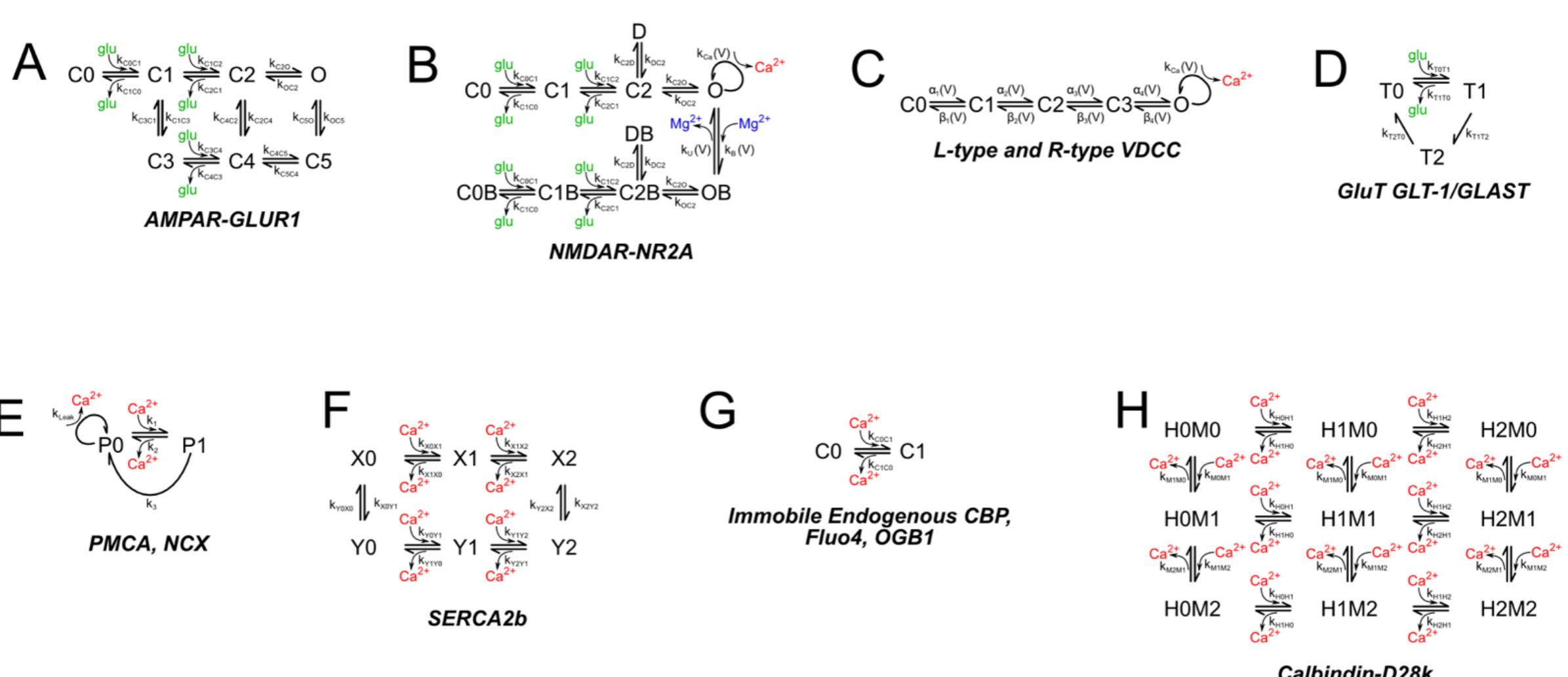
Realistic Model of Synaptic Signaling in Neuropil

Realistic synaptic morphology from ssTEM reconstruction of hippocampal neuropil (collaboration with Kristen Harris, UT Austin)



Model Constraints

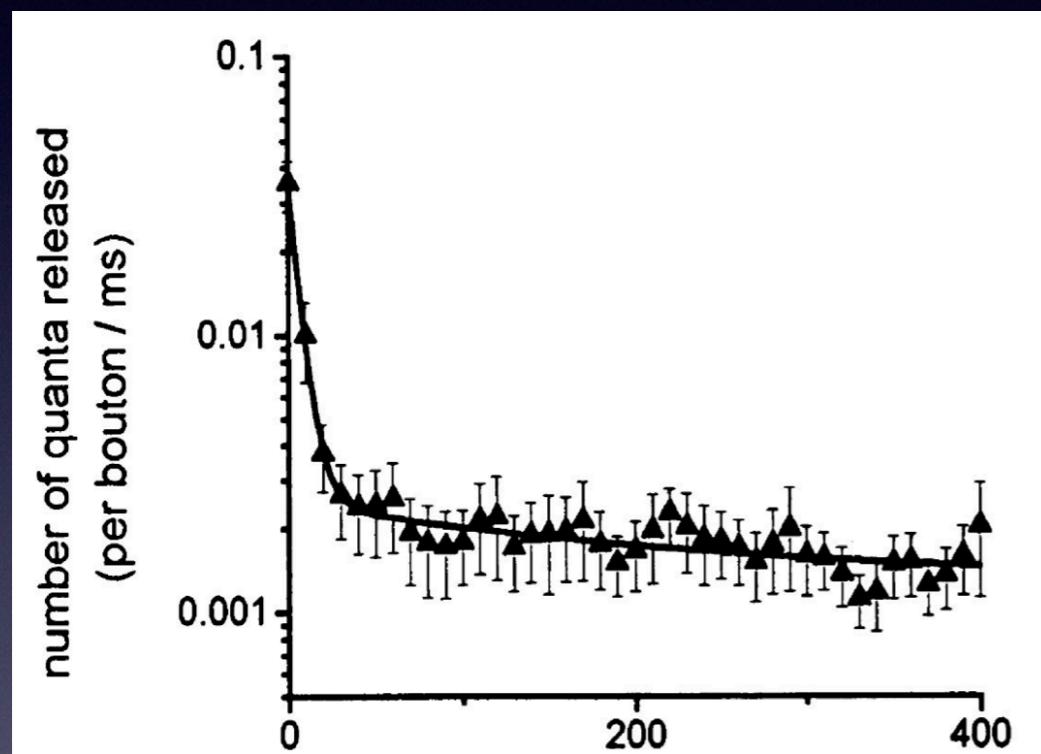
Experimental estimates of reaction kinetics and subcellular distributions for important molecules
(collaboration with Mary Kennedy, CalTech)



Model Constraints

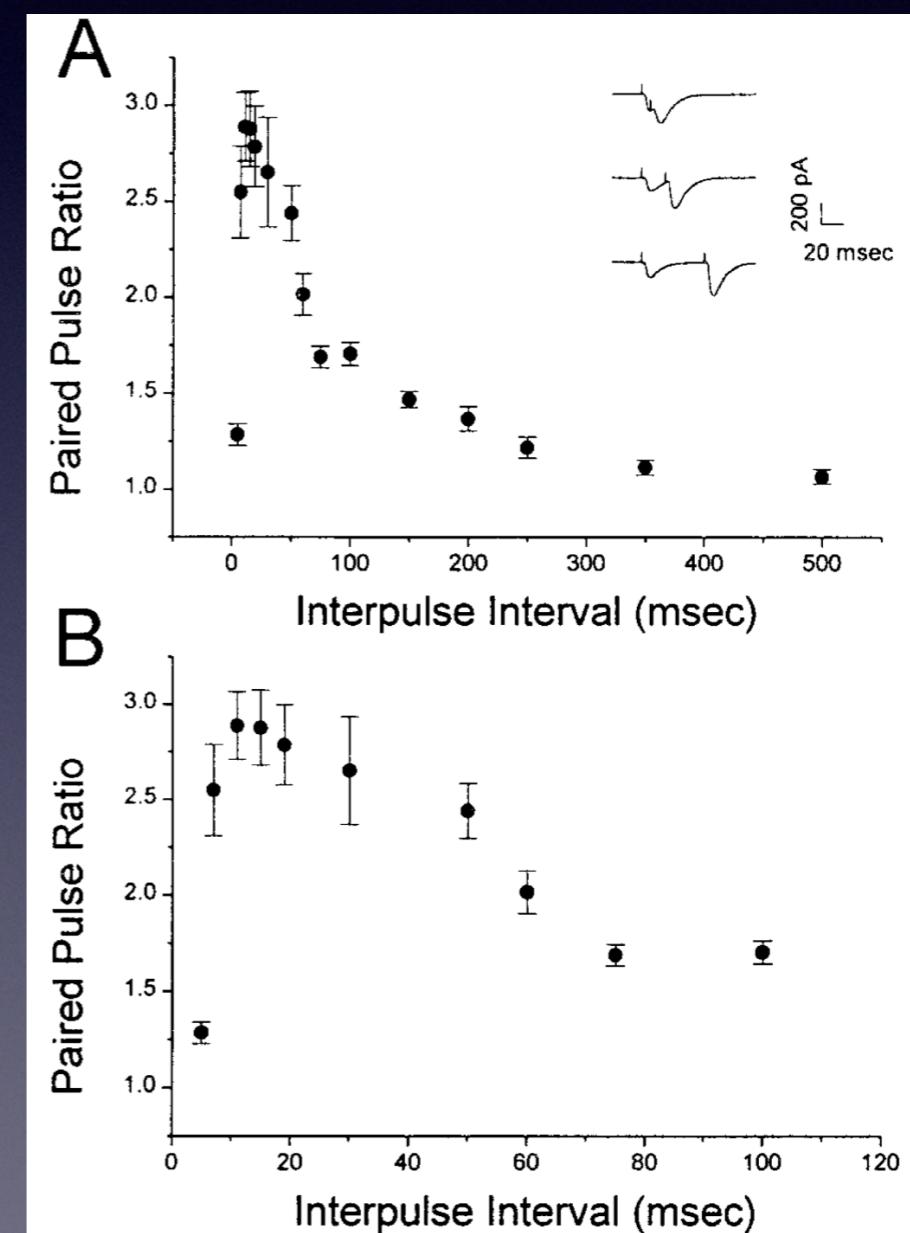
Experimental observations of evoked vesicular release:
(collaboration with Suhita Nadkarni, ISER Pune)

single stimulus



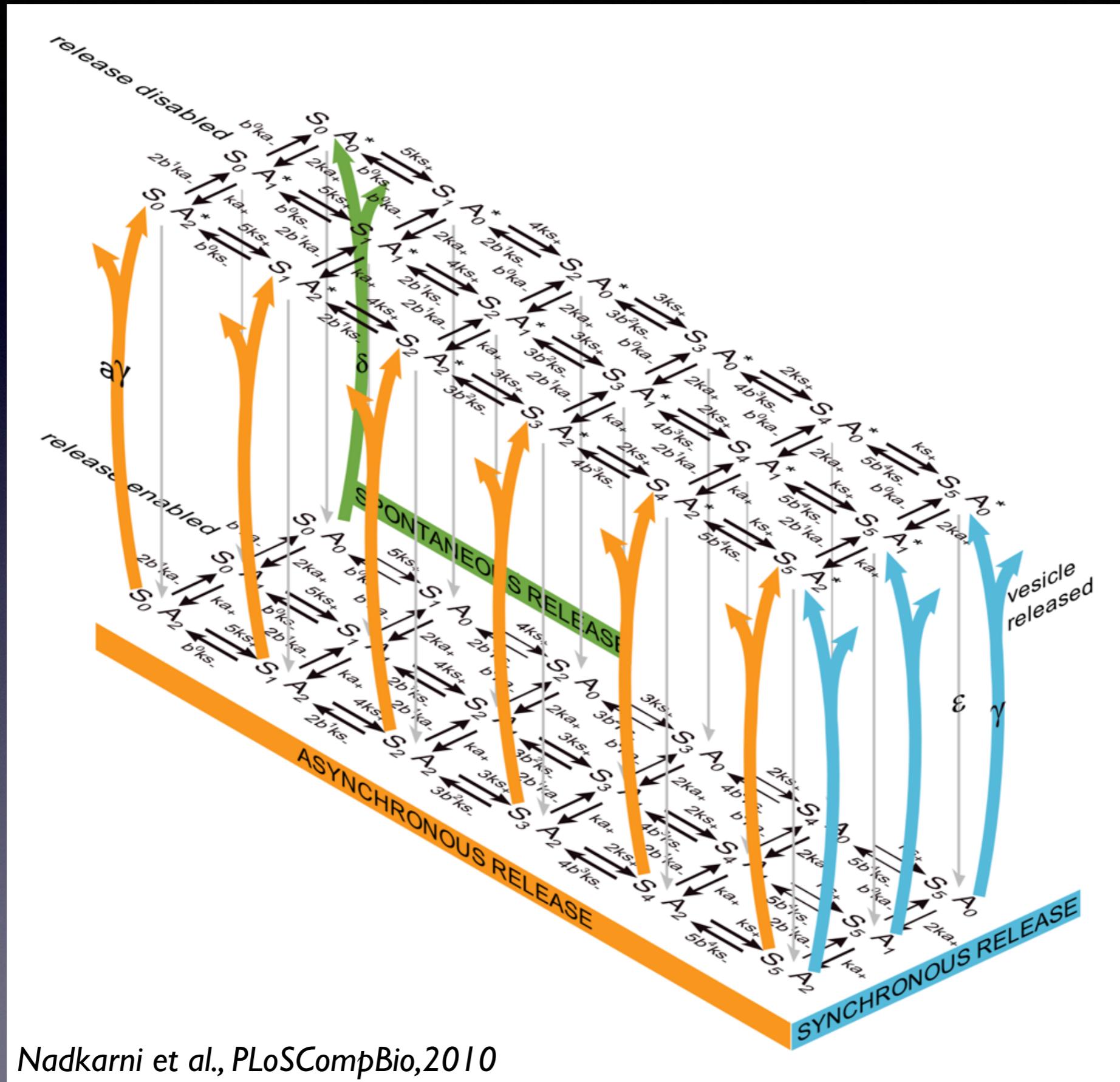
Goda & Stevens, PNAS, 1994

paired-pulse stimulus

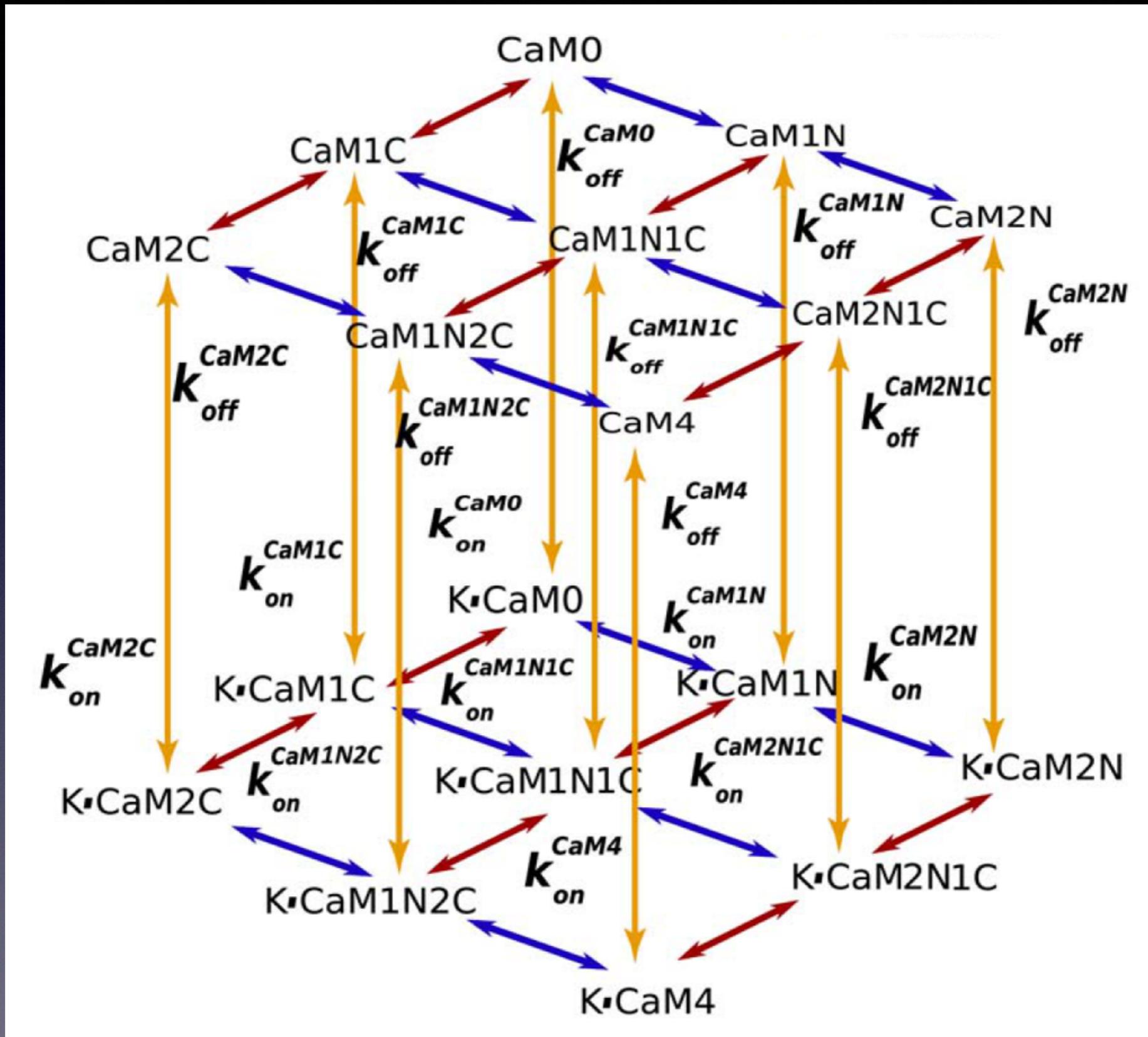


Dobrunz, Huang & Stevens, PNAS, 1997

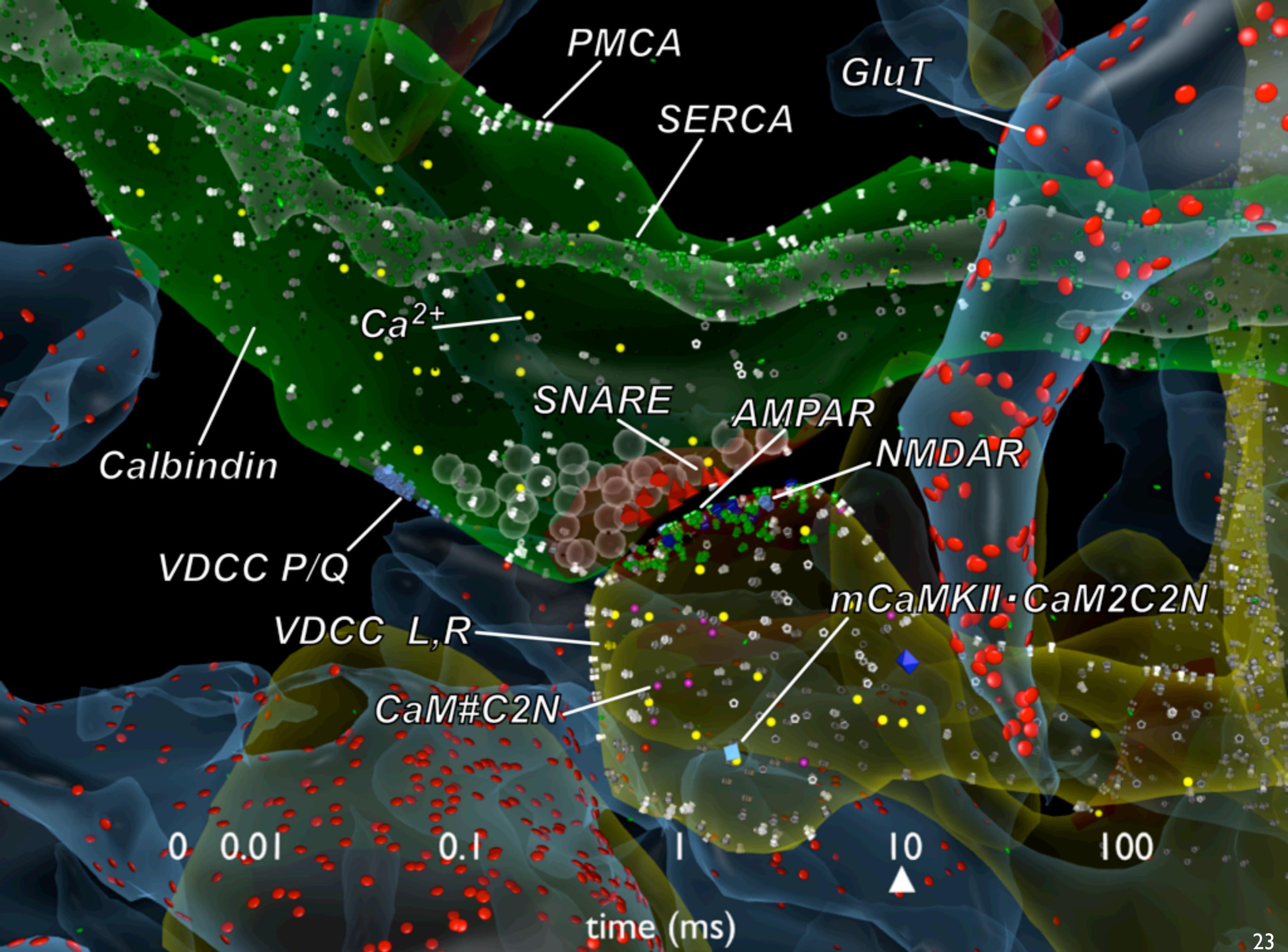
Model Constraint: kinetics of calcium sensor in SNARE complex

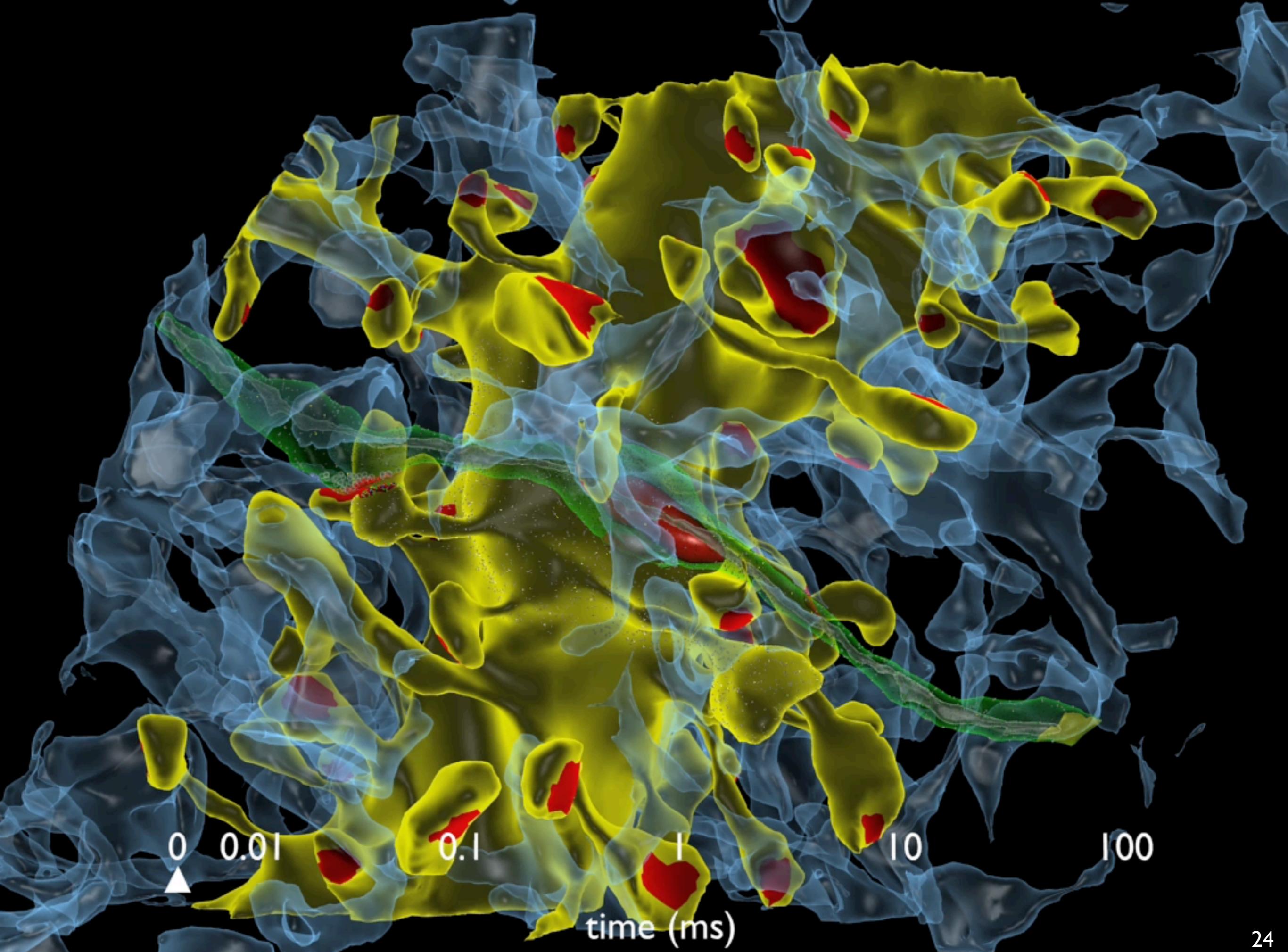


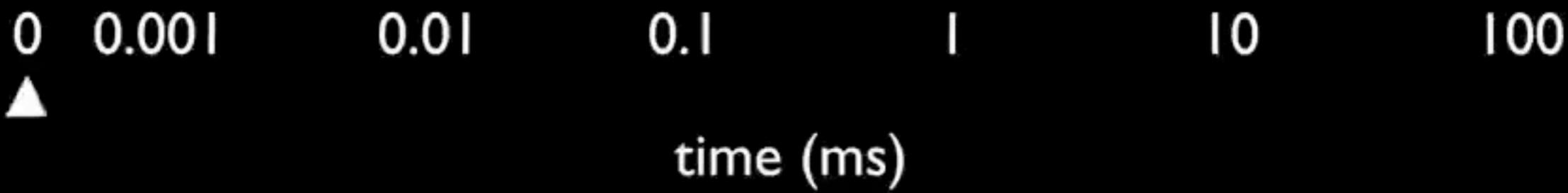
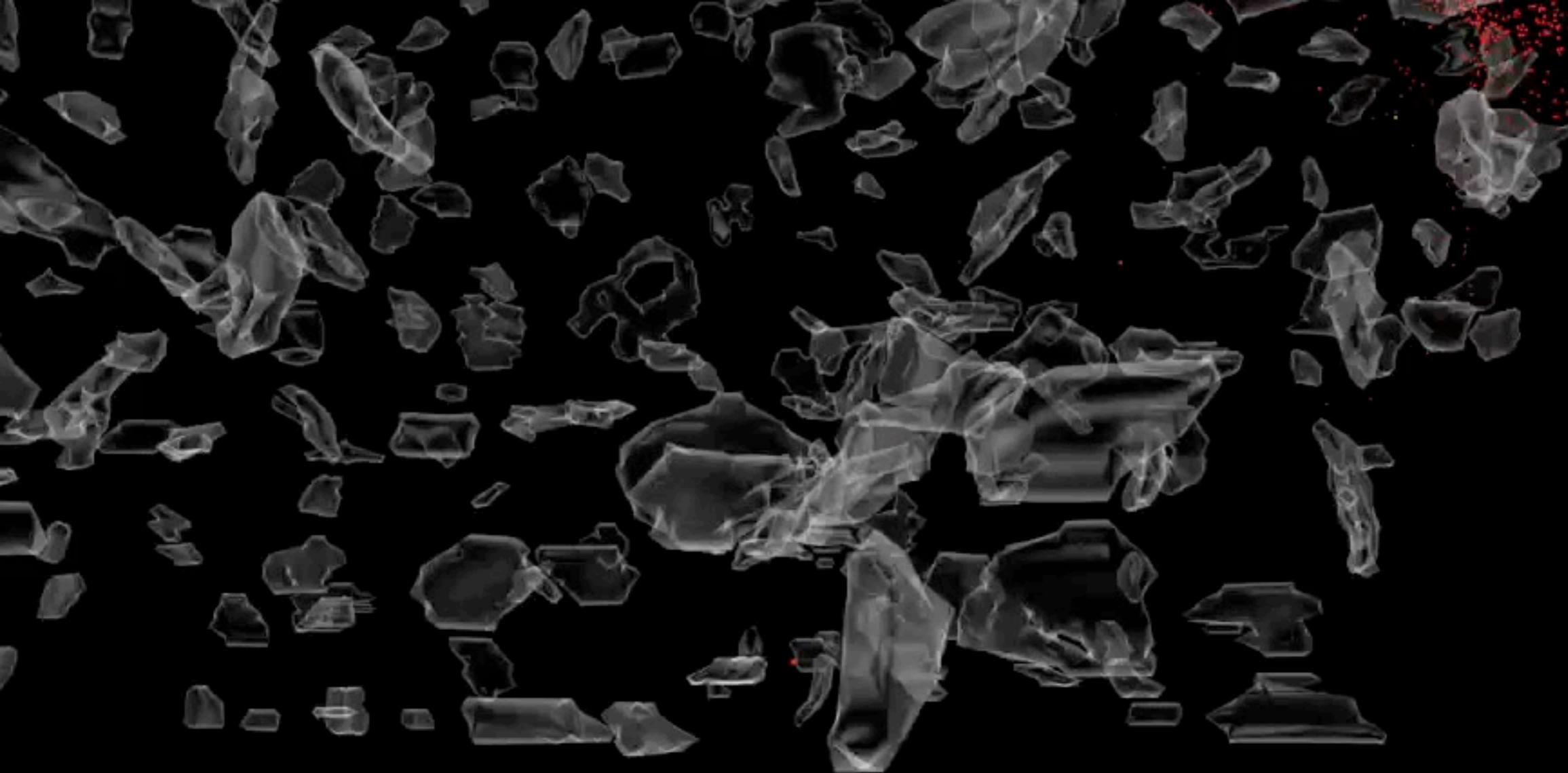
Model Constraint: kinetics of calmodulin and CaMKII



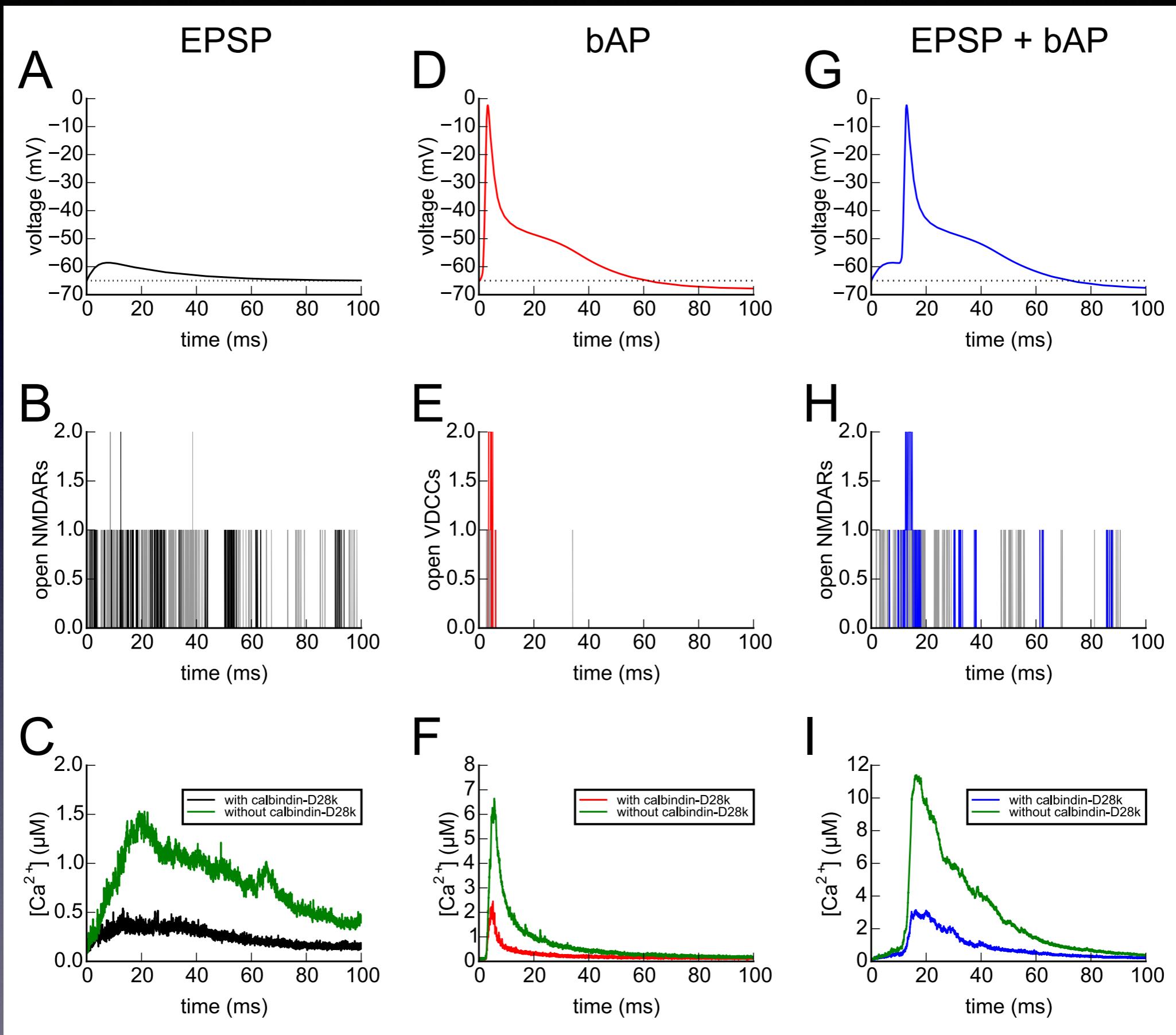
Mary Kennedy (Pepke et al., PLoS CompBio, 2010)



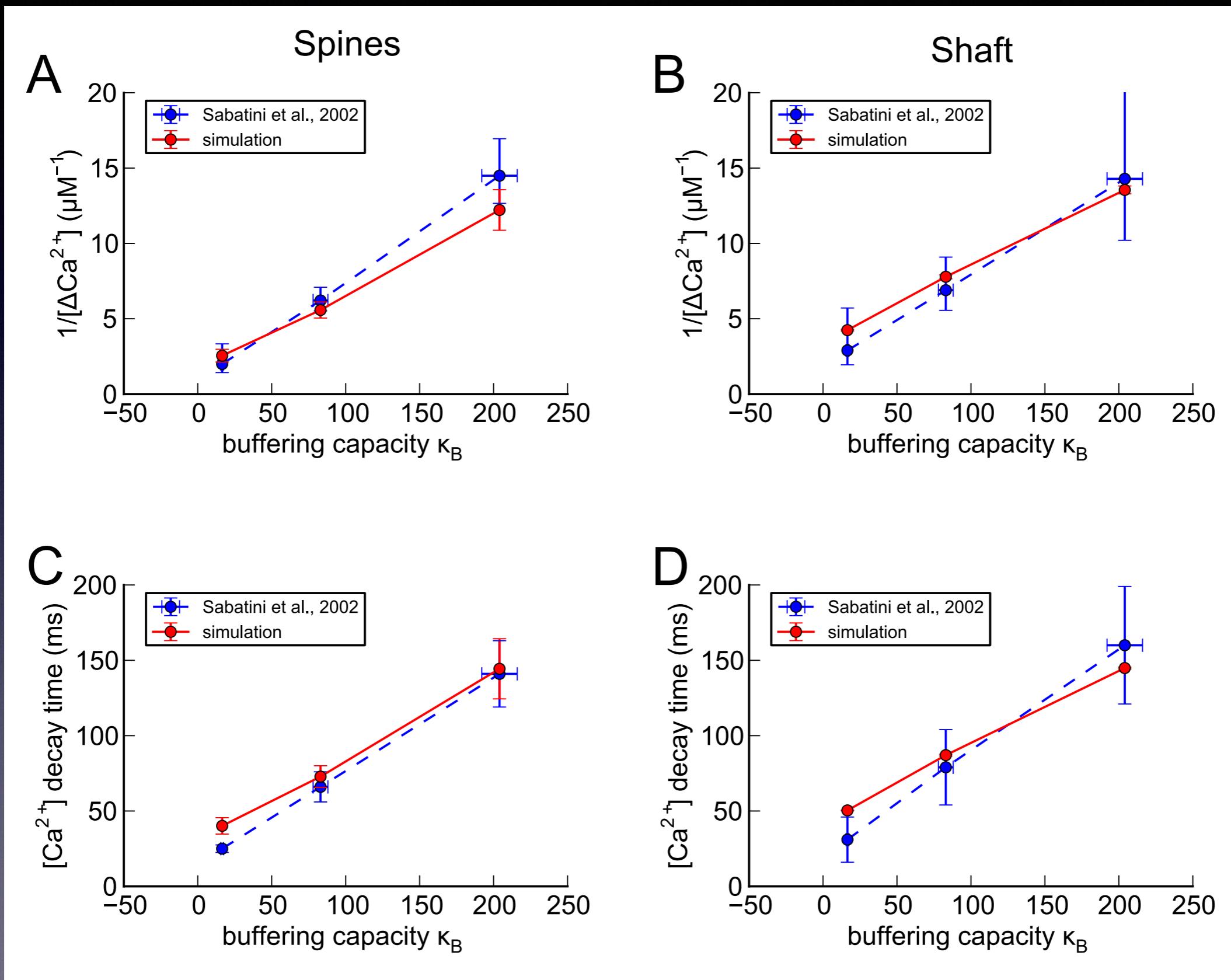




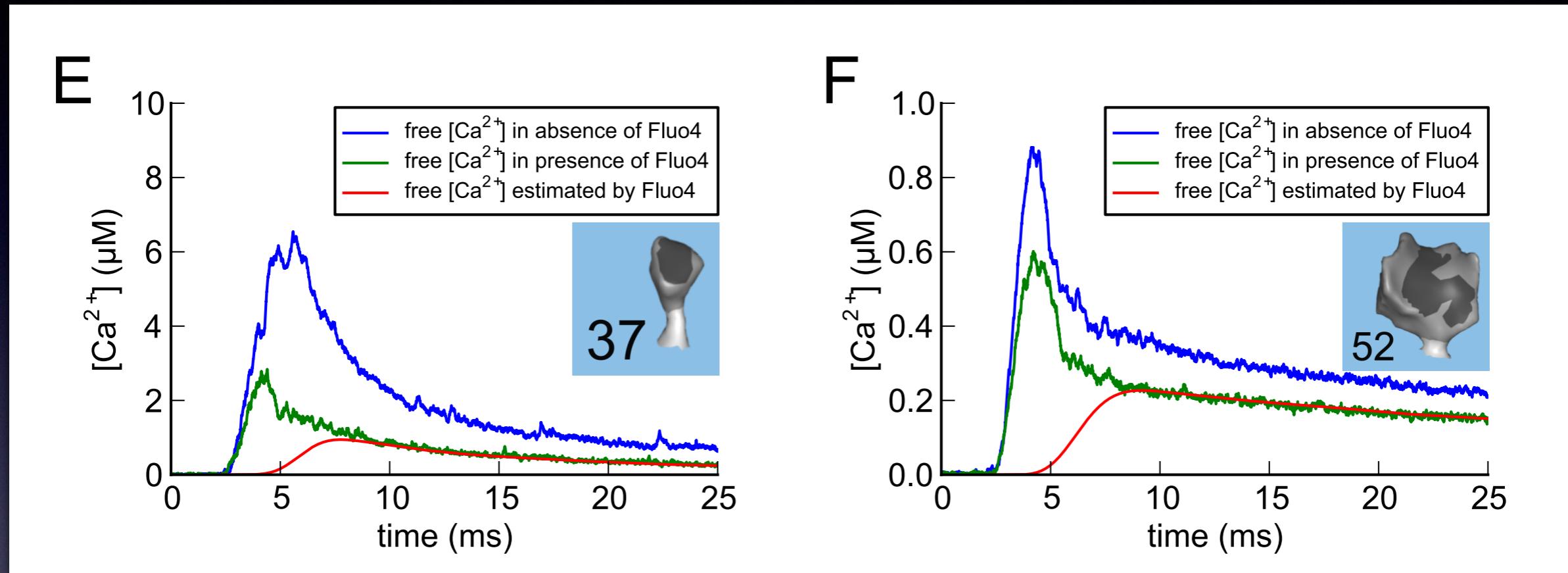
Predicted Synaptic Behavior



Model Constraint: Calcium Buffering Capacity



Predicted Impact of Exogenous Ca^{2+} Indicators



Bartol et al., *Frontiers Synaptic NeuroSci*, 2015

Acknowledgements

Salk

Terry Sejnowski
Bob Kuczewski
Adam Husar
Joel Yancey
Jed Wing
Rex Kerr
Ben Regner
Jay Coggan
Courtney Lopreore
Dan Keller
Justin Kinney
Kevin Franks
Cailey Bromer
Oliver Ernst
Jonathan Garcia
Eugene Kim
Elaine Zhang
Mariam Ordyan
Guadalupe Garcia
Sara Sameni
Mohamad Samavat
Margot Wagner
Rachel Mendelsohn
Don Spencer

ISER Pune
Suhita Nadkarni

PSC

Markus Dittrich
Jacob Czech
Art Wetzel
Greg Hood
Rozita Laghaei

Caltech

Mary Kennedy
Melanie Stephan
Shirley Pepke

UT Austin

Kristen Harris
Chandra Bajaj

Univ. Bordeaux

Daniel Choquet
Eric Hosy
Olivier Thoumine
Kalina Haas
Benjamin Compans
Mathieu Letellier

UPitt

Ivet Bahar
James Faeder
Ali Sinan Saglam
Jose-Juan Tapia
Dipak Barua
Sanjana Gupta
Cihan Kaya

CMU

Bob Murphy
Devin Sullivan
Ivan Cao-Berg

UCSD

Mark Ellisman
Maryann Martone
Rommie Amaro
Padmini Rangamani
Michael Holst
Sophia Hirakis
Christopher Lee
John Moody
Jeff Bush

NIH & NSF

MCell Models as Python Programs

Model Export from CellBlender generates a well-structured Python program

- model.py # the main routine of the model
- parameters.py # the parameters, of course
- subsystem.py # molecule types and reaction rules
- model.bngl # representation of model in BNGL
- geometry module # the triangulated meshes
- instantiation.py # mapping of species into geometry
- observables.py # numeric and visualization observables
- **customization.py # user-defined customizations**

CellBlender Export -> data_model.json
data_model_to_pymcell -b data_model.json -> (model.py etc...)

MCell Python can also generate a data_model.json file

CellBlender can import a data_model.json file

CellBlender can import a compartmental BNGL file

Overview of MCell4 Callbacks

Callbacks provide hooks into MCell4's event-driven physics engine

Provide a means to customize the biophysics in a model beyond the built-in capabilities of MCell4, including coupling to external physics engines!!!

A callback is defined by associating (registering) a custom python function with an event that occurs during the simulation.

Types of Callbacks:

- Mol-Wall collision callback
- Reaction callback
- Iteration “callback” - - call user code inside iteration loop