# A subcolumn sampler for driving subgrid-scale microphysics

Vincent Larson

DWD Seminar

25 Feb 2016

# Outline

- The problem: how to drive microphysics using subgrid variability in clouds
- One approach to the problem: deterministic quadrature
- Another approach to the problem: Monte Carlo integration
- Reducing noise in Monte Carlo: SILHS

# The Reynolds-averaged equations are *integro*-differential equations:

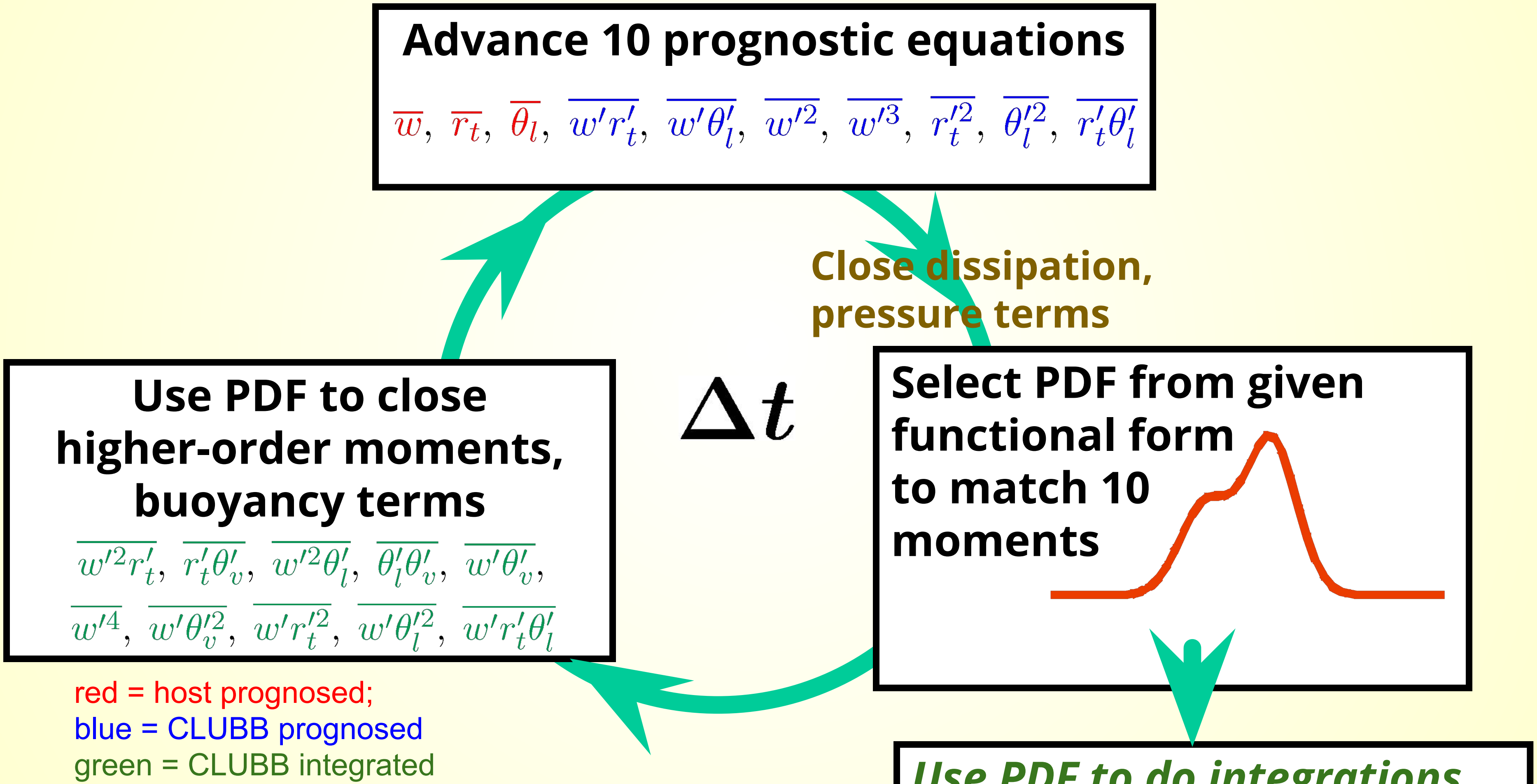
$$\frac{\partial \overline{r_t}}{\partial t} = \underbrace{-\overline{w}\frac{\partial \overline{r_t}}{\partial z}}_{ma} \underbrace{-\frac{\partial \overline{w' r_t'}}{\partial z}}_{ta} + \overline{\mathrm{Mphys}}$$

$$\frac{\partial \overline{\theta_l}}{\partial t} = \underbrace{-\overline{w}\frac{\partial \overline{\theta_l}}{\partial z}}_{ma} \underbrace{-\frac{\partial \overline{w' \theta_l'}}{\partial z}}_{ta} + \overline{\mathrm{RT}} + \overline{\mathrm{Mphys}}$$

$r_t$ = total water
*theta*$_l$ = liq water pot temp
*w* = vertical velocity

red = predicted by host model, microphysics (Mphys), or radiation (RT)
blue = predicted by cloud/turbulence parameterization
green = subgrid integration driven by cloud parameterization

# How CLUBB advances the solution one timestep:




Golaz et al. (2002)

Golaz et al. (2002)

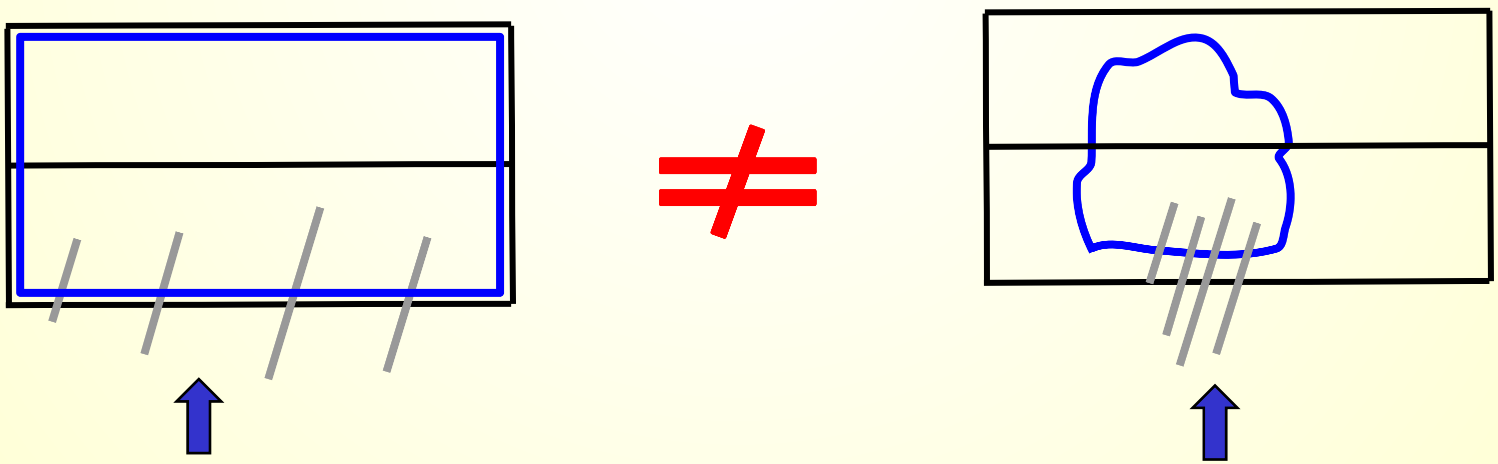
# The problem, in words:

If we don’t account for subgrid-scale spatial variability in a large-scale numerical model, then errors may result because of the non-linearity of some physical processes, such as microphysics (e.g. autoconversion).

# The problem, in pictures:

We'd like to drive microphysical processes using subgrid-scale variability.

For instance, we'd like to account for the effects of partial cloudiness on drizzle rate.  We'd also like to account for within-cloud variability.



How a model with no subgrid information handles microphysics

What we are attempting to do

# The problem, in an equation:

Consider the process of drizzle formation ("autoconversion" *A*). Because autoconversion is *nonlinear*,

$$A(\overline{q_l}) \neq \overline{A(q_l)}$$

What a model predicts if it ignores subgrid variability.

What we really want to parameterize.

$A$ = Autoconversion: cloud to drizzle
$q_l$ = Liquid cloud water mixing ratio
Overbar = Grid box average

Pincus and Klein (2000)

# How can we fix the averaging errors?

By integration. We could remove the averaging errors if we could predict the PDF of relevant subgrid variability for each grid box and time step. Then the problem is reduced to integration.

$$\overline{A(q_l)} = \int P(q_l) A(q_l) \, dq_l$$

Grid box avg autoconversion

Probability density function (PDF)

``Local" value of autoconversion

Larson et al. (2002)

## What we are trying to do about it: Develop methods to integrate subgrid variability

- An ideal method would include at least 4 desired features . . .

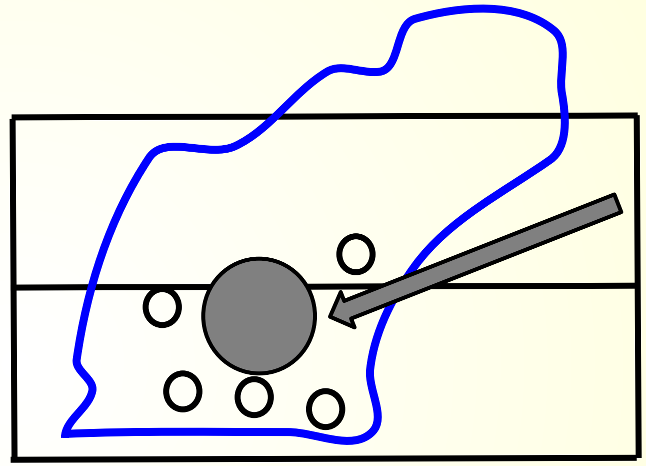
# Desired feature 1: Ability to handle *horizontal correlations* between variates

The parameterization should include *co-variability* of all relevant fields, including cloud fields, but also turbulence and hydrometeor fields.

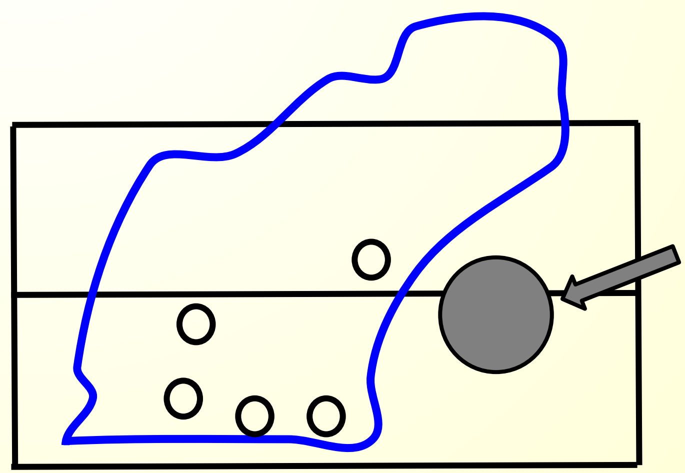
# For instance, collection of cloud drops by rain depends on the covariability of cloud and rain

Some microphysical processes involve two or more hydrometeor species. An example is collection of cloud droplets by rain drops.

Such processes depend on the correlation of multiple species.



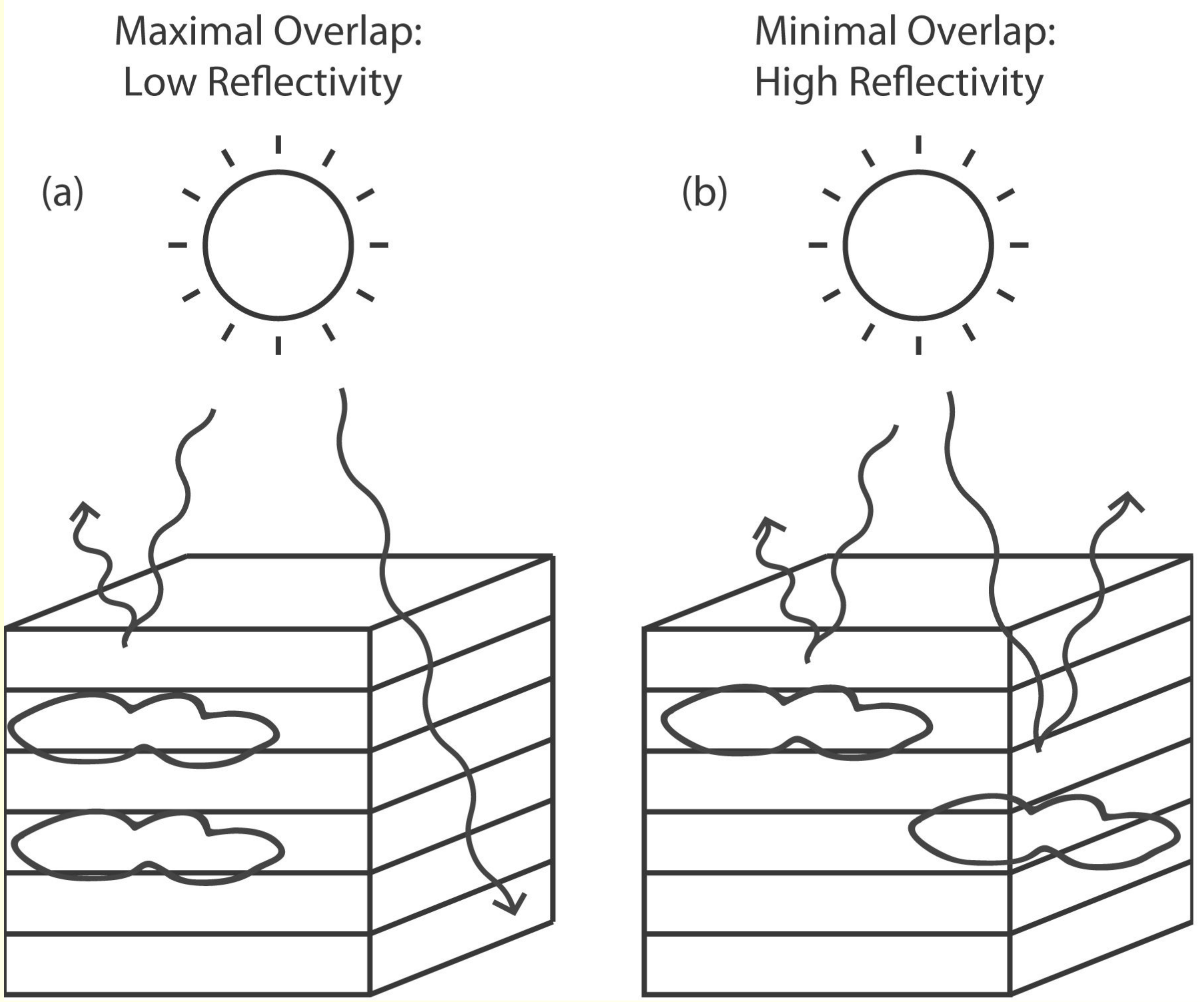
This rain drop will collect many cloud drops:



This rain drop will not:

# Desired feature 2: Ability to handle *vertical* correlations

Vertical correlations of clouds (a.k.a. “cloud overlap”) is important for computing cloud albedo in climate models. (e.g. Liang and Wang, 1997)



# Desired feature 3: *Consistency* of the subgrid representation among physics parameterizations

The parameterization of subgrid variability should feed the same subgrid fields into various microphysical calculations.

# Desired feature 4: Modularity of software

Ideally, the parameterization of subgrid variability should be kept separate from the physics parameterizations.

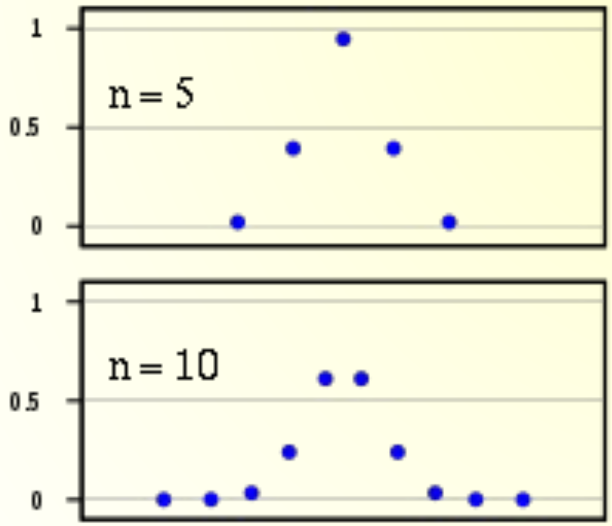
Otherwise, we would need to update the subgrid variability code whenever the microphysics changes.

# Outline

- The problem: how to drive microphysics using subgrid variability in clouds
- One approach to the problem: deterministic quadrature
- Another approach to the problem: Monte Carlo integration
- Reducing noise in Monte Carlo: SILHS

# One approach is to use Gaussian quadrature

This approach is highly accurate for smooth integrands and can be extended to lognormal PDFs.




It works by evaluating the function at special quadrature points that produce exact answers for low-order polynomials.

Chowdhary et al. (2015, MWR)

# Unlike analytic integration, deterministic quadrature can integrate even complicated microphysical formulas

Suppose ice microphysics is parameterized by a complicated numerical subroutine.

Analytic integration is impossible, but deterministic quadrature is feasible.

# Deterministic quadrature is more accurate than Monte Carlo for smooth functions, but not discontinuous ones



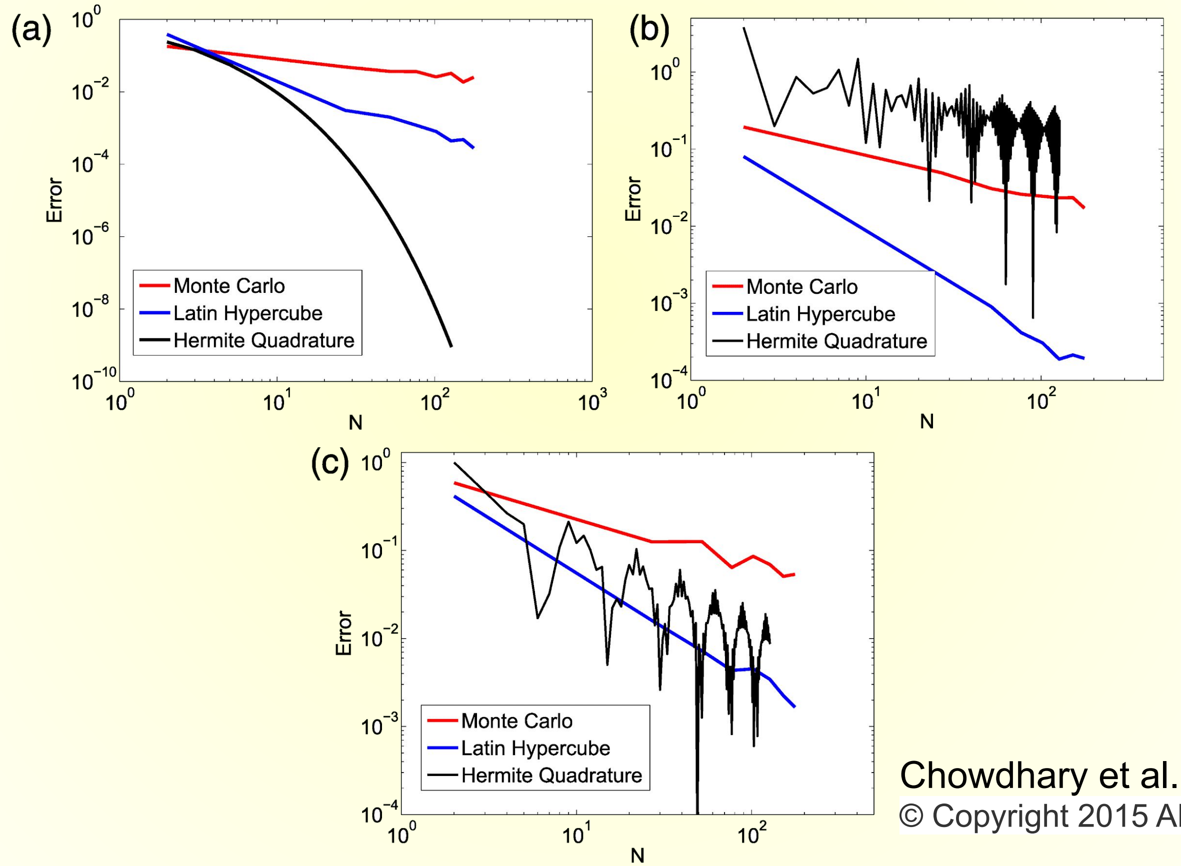



FIG. 1. (a) Hermite quadrature easily outperforms both Monte Carlo and Latin hypercube sampling approaches. (b) Hermite quadrature exhibits very poor performance when we add a single jump discontinuity at $a = 1$ where $f(x) = 1/(1 + x^2)$. (c) Convergence for Hermite quadrature with $f(x) = |x - 1|$, where we integrate over the entire domain. This time, the discontinuity is present only in the derivative so convergence is slightly better.

Chowdhary et al. (2015, MWR)

# Deterministic quadrature can be used with a variety of PDF shapes that cover different portions of the domain



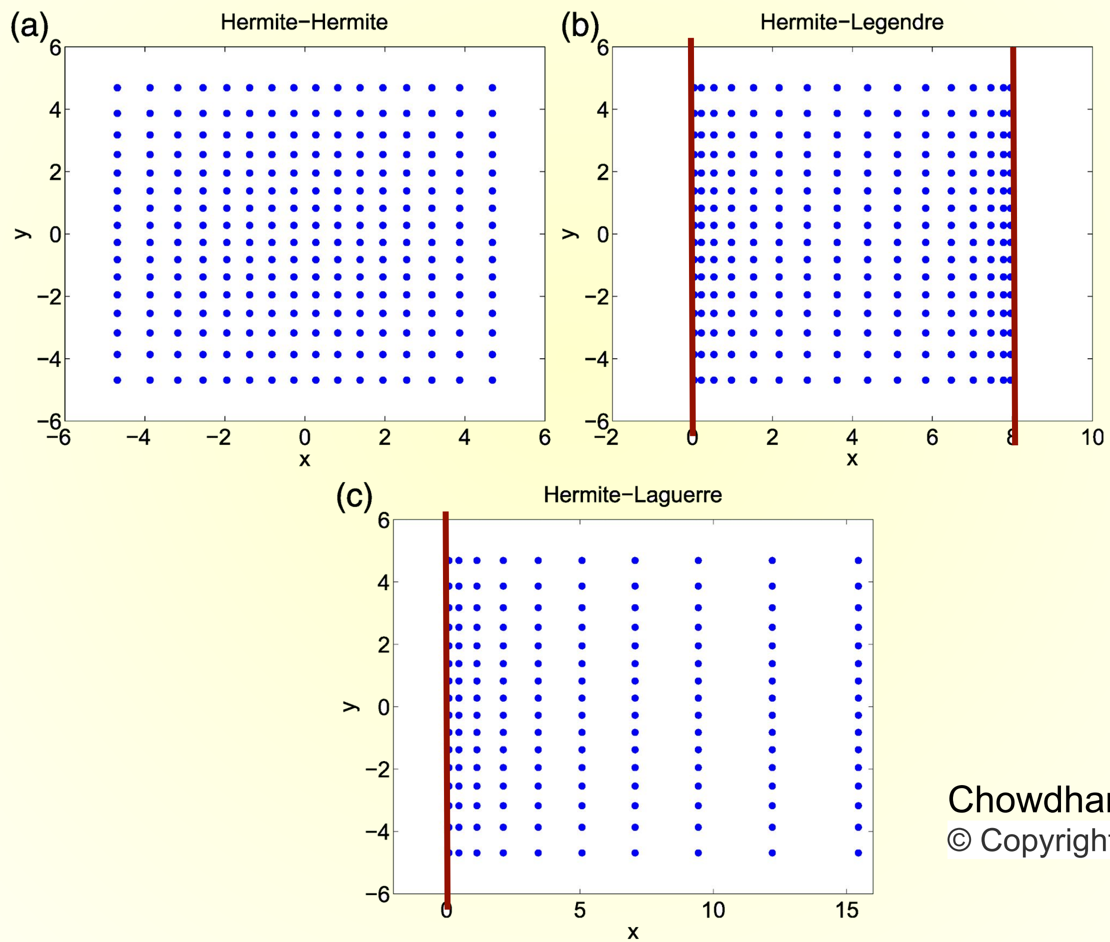



FIG. 3. (left to right), (top to bottom) Two-dimensional tensor product quadrature rules using Hermite–Hermite, Hermite–Legendre, and Hermite–Laguerre in the *x* and *y* directions, respectively.

Chowdhary et al. (2015, MWR)

# Deterministic quadrature is promising, but the required software changes are somewhat intrusive

Monte Carlo simulation can handle high-dimensional integrals, and hence it can treat the entire microphysics as a black box.

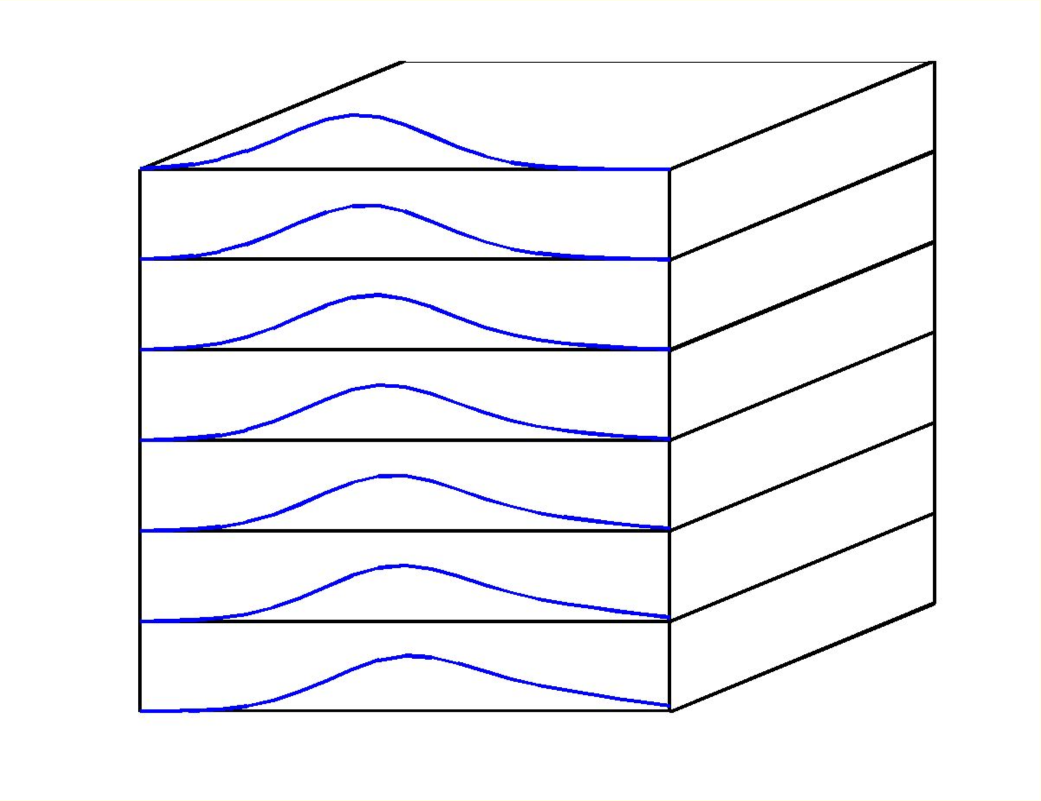
Deterministic quadrature is better in lower dimensions, and hence it needs to integrate each microphysical process individually.

# Outline

- The problem: how to drive microphysics using subgrid variability in clouds
- One approach to the problem: deterministic quadrature
- Another approach to the problem: Monte Carlo integration
- Reducing noise in Monte Carlo: SILHS

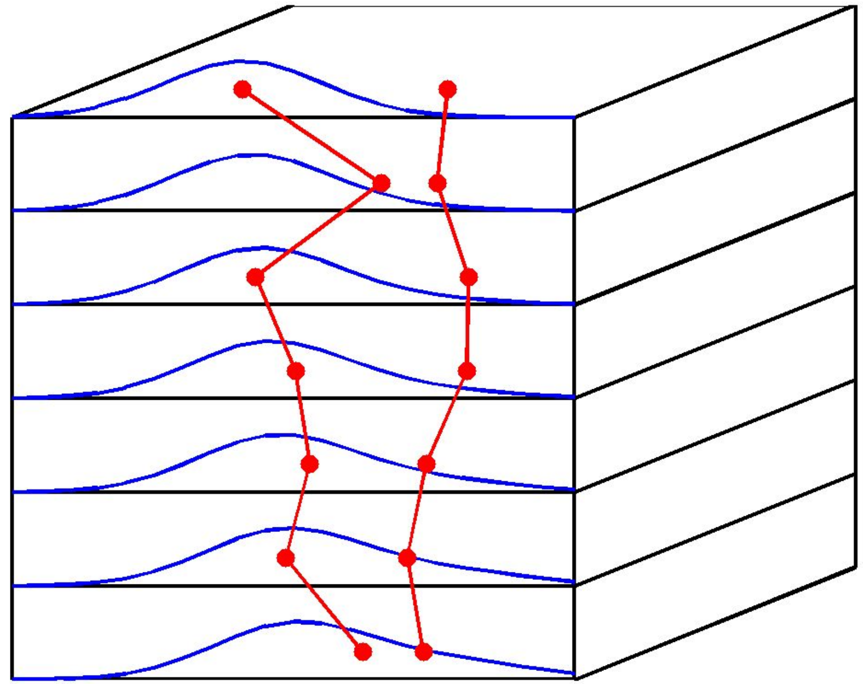
# The 4-step method we are developing to parameterize subgrid variability and drive microphysics:

1. Predict the probability density function (PDF) of subgrid variability at each grid level.



# The 4-step method we are developing to parameterize subgrid variability and drive microphysics:

2. Generate subcolumns that are consistent with the subgrid PDF at each level and satisfy a suitable overlap assumption.



# The 4-step method we are developing to parameterize subgrid variability and drive microphysics:

3. Feed subcolumns, one by one, into the microphysics parameterization.

4. Average the microphysics tendencies from subcolumns and feed them back into the large-scale (host) model.

# To do the integrals, we can use Monte Carlo integration

We can approximate the grid box average using Monte Carlo integration. That is, we sample $P(q_l)$ randomly, substitute these values into $A(q_l)$, and compute $\overline{A(q_l)}$ as a typical statistical average.

We can choose a small number of sample points per grid box and time step. Over many time steps, an unbiased average will emerge. However, this procedure introduces statistical noise into the simulation.

Barker et al. (2002), Pincus et al. (2005)

## To handle vertical overlap, we assume samples are correlated in the vertical

- The sample points in a subcolumn are chosen at successive vertical levels.
- The correlation between vertical points is assumed to fall off exponentially with vertical distance between them.
- The main purpose is to prepare the way for treating radiative transfer through clouds.

# Monte Carlo integration is non-intrusive

Monte Carlo integration acts as an *interface* between the host model and a microphysical parameterization.

It allows us to update a microphysical parameterizations without updating the integration method.

Pincus et al. (2003)

# A key problem with Monte Carlo is computational cost

Doubling the number of sample points doubles the cost of the microphysics.

We can only afford a few sample points per grid box and time step.

However, the microphysical calculations for each sample point are independent and hence parallelizable.

# A comparison of deterministic quadrature and Monte Carlo integration

| | Deterministic quadrature | Monte Carlo |
|---|---|---|
| **Computationally efficient** | Yes | No |
| **Modular** | No | Yes |
| **Includes vertical correlations?** | No | Yes |
| **Can be used for arbitrary PDF?** | No | Yes |
| **Can be used for arbitrary physics function?** | Yes | Yes |
| **Includes horizontal correlations?** | Yes | Yes |
| **Promotes consistency?** | Yes | Yes |

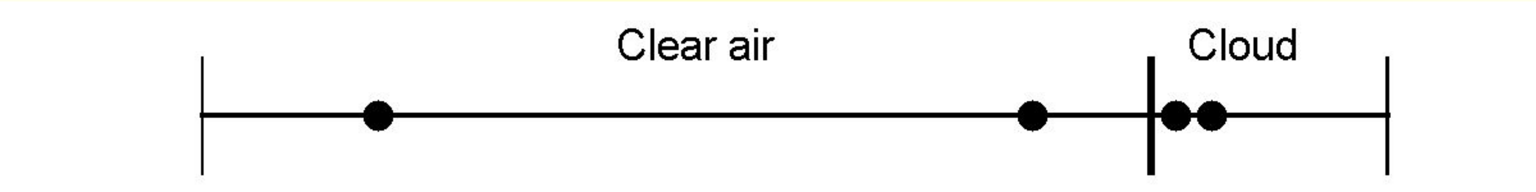
# Outline

- The problem: how to drive microphysics using subgrid variability in clouds
- One approach to the problem: deterministic quadrature
- Another approach to the problem: Monte Carlo integration
- Reducing noise in Monte Carlo: SILHS

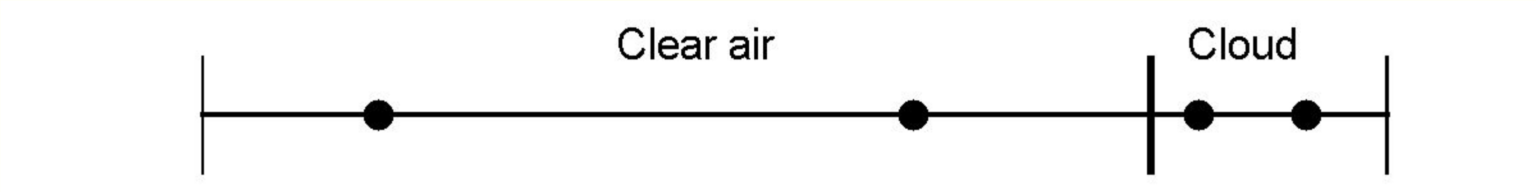
# We have developed a Monte Carlo sampler: SILHS (Subgrid Importance Latin Hypercube Sampler)

- SILHS is a hybrid between an importance sampler and a stratified (Latin hypercube) sampler.

- Importance sampling: We preferentially sample an "important" region, namely cloud:




- Stratified sampling: Within important regions, we stratify (i.e. spread out) the sample points, so that they don't clump.



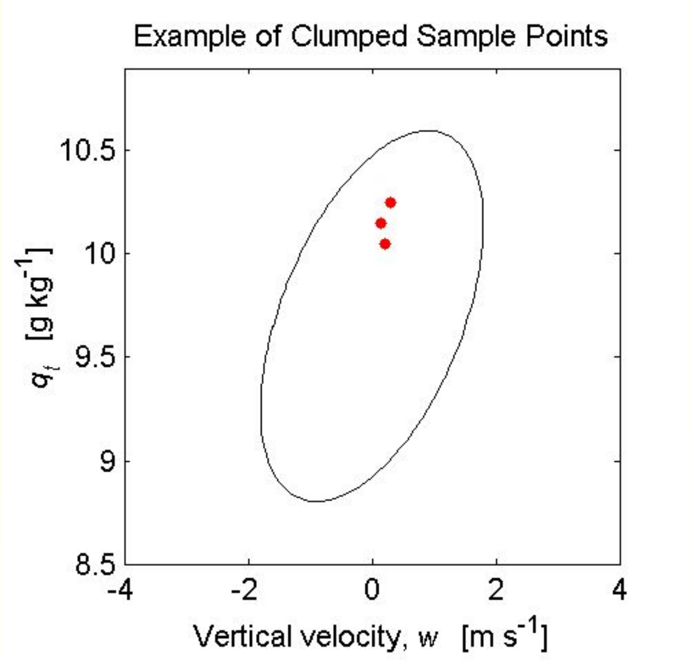

Larson and Schanen (2013)

# How can we reduce noise in Monte Carlo Integration?

To stratify the sample points, we may use

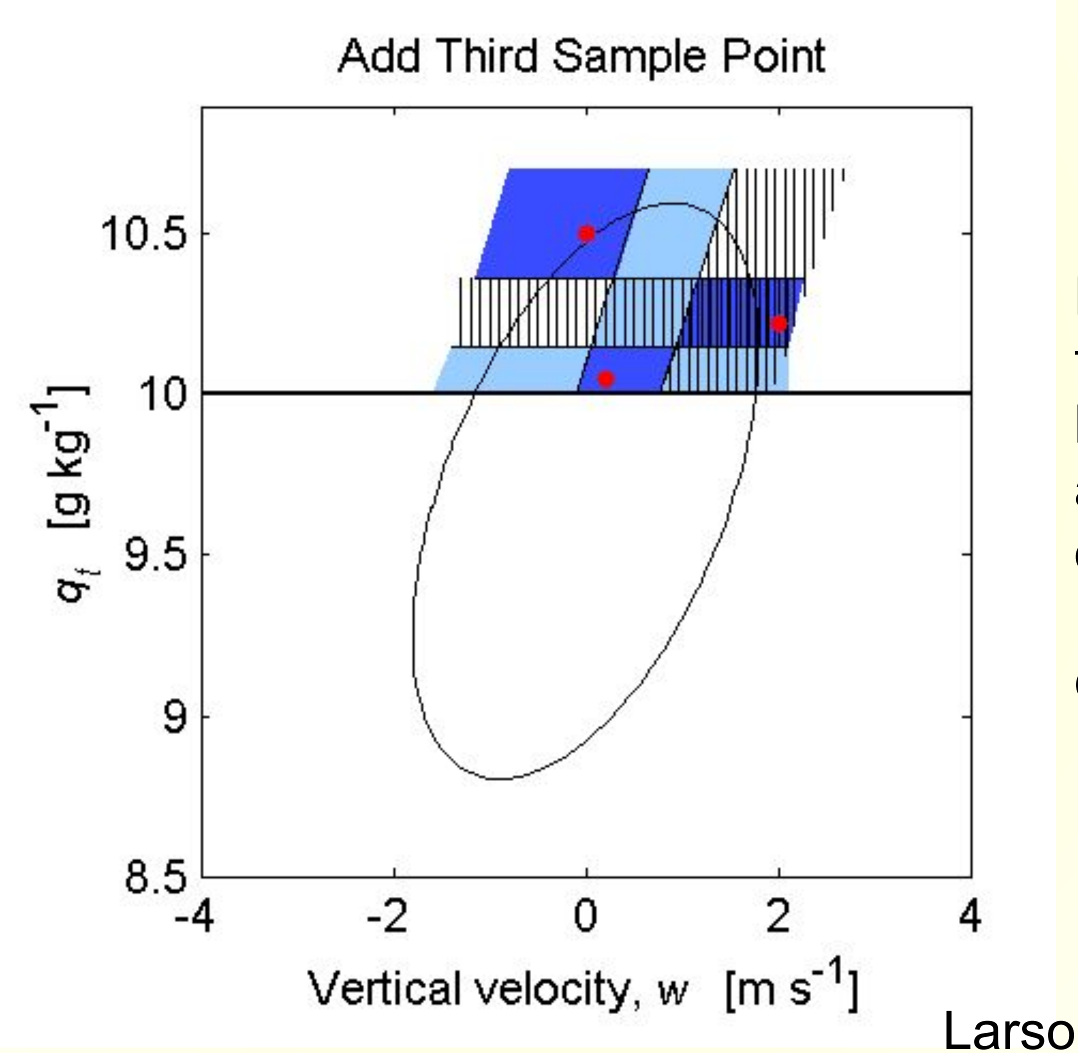
Latin Hypercube Sampling.

This is a *type of Monte Carlo Sampling* that spreads out the sample points, so that they don't clump together, as can happen with straightforward Monte Carlo Sampling.




McKay, Beckman, Conover (1979)

# Latin Hypercube Algorithm




LH guarantees that the sample has low, med, and high values of both $w$ and $q_t$. I.e. points do not cluster.

Larson et al. (2005)

# SILHS' upgraded importance sampling method

- The goal is to make sure all important processes are well sampled
- Original importance sampling targeted cloudy region of grid box
  - An out-of-cloud process, evaporation of rain, was ignored
- New method divides grid box into different regions, or "categories"
  - Categories based on cloud, precipitation, and PDF mixture component
  - Up to 8 categories can be used
- The sampling density can be adjusted individually for each category

# Different processes act in different regions

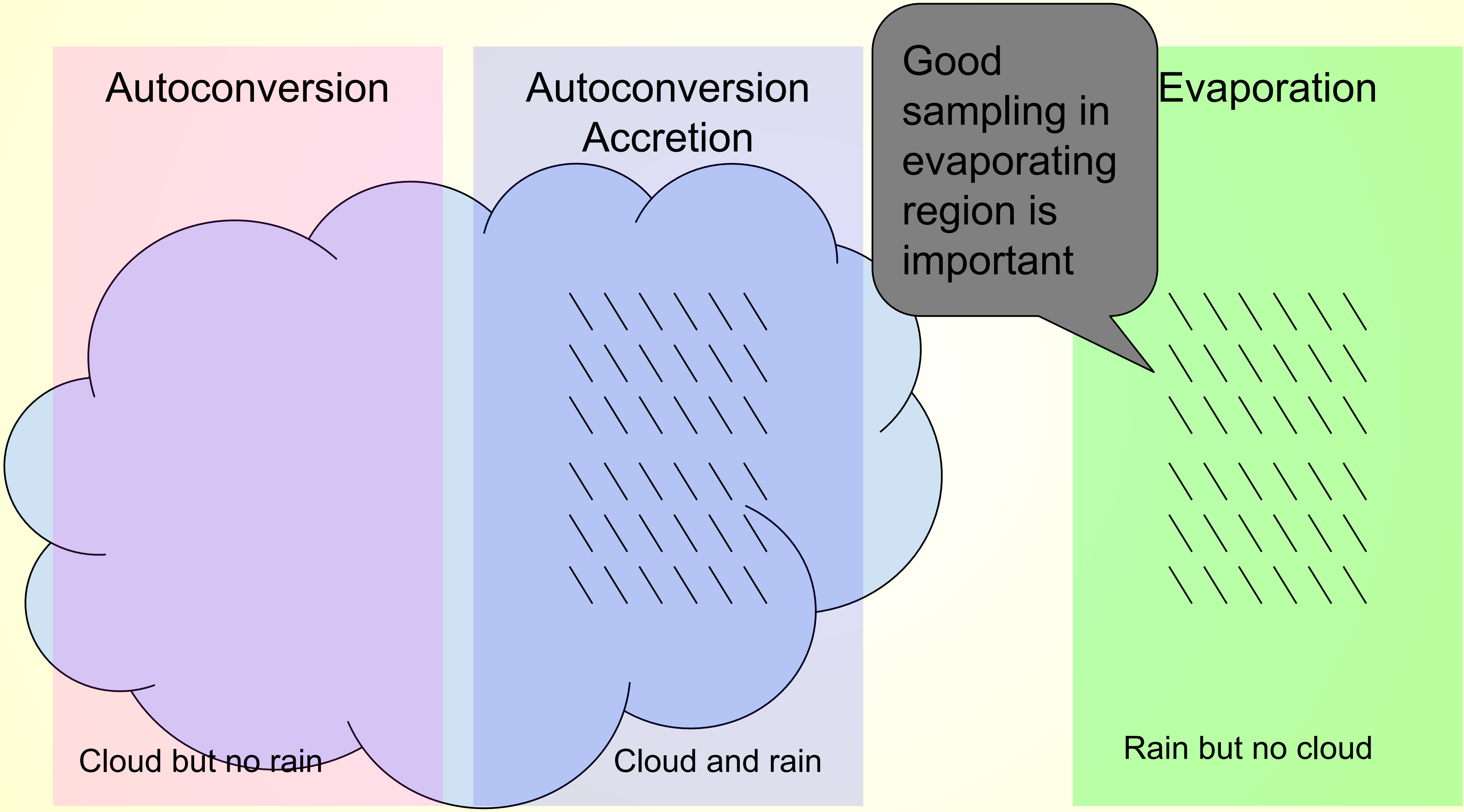



Figure credit: Eric Raut

# CLUBB's PDF consists of distinct categories

Each of the 2 mixture components of the PDF has a mixture fraction $\xi_{(1,2)}$, a precipitation fraction $f_{\mathrm{p}(1,2)}$, and a cloudy fraction. The PDF is:

$$P(\vec{x}) = \sum_{m=1}^{2} \xi_{(m)} \left[ f_{\mathrm{p}(m)} P_{(m)}(r_t, \theta_l, w, N_{\mathsf{c}}, \mathbf{hm}) \right.$$
$$\left. + (1 - f_{\mathrm{p}(m)}) \delta(\mathbf{hm}) P_{(m)}(r_t, \theta_l, w, N_{\mathsf{c}}) \right] .$$

Here $\mathbf{hm}$ denotes the vector of hydrometeors.

Raut and Larson (2016, GMD)

# Some categories are more important than others. So let's vary the sample densities. This is importance sampling.

Each category $C_j$ is associated with a certain amount of PDF mass, $p_j$:

$$p_j = \int_{C_j} P(\vec{x}) \mathrm{d}\vec{x}.$$

To preferentially sample the "important" categories, we sample category, $C_j$, with a user-defined probability, $S_j$, rather than $p_j$.

# To implement this, we create a new PDF ("*Q*") that reflects the desired sampling densities

We define a new function, $L(\vec{x})$, called the "likelihood ratio":

$$L(\vec{x}) \equiv \sum_{j=1}^{N_{\text{cat}}} \left( \frac{p_j}{S_j} \right) \cdot 1_j(\vec{x}).$$

Here, $1_j(\vec{x})$ is the indicator function of category $C_j$, defined as

$$1_j(\vec{x}) = \begin{cases} 1 & \vec{x} \in C_j \\ 0 & \vec{x} \notin C_j \end{cases},$$

The new sampling PDF is defined as

$$Q(\vec{x}) = \frac{P(\vec{x})}{L(\vec{x})}.$$

# We draw samples from *Q* and feed them into a modified integrand

The integral is re-written as

$$\int A(\vec{x})P(\vec{x})\mathrm{d}\vec{x} = \int A(\vec{x})L(\vec{x})\left(\frac{P(\vec{x})}{L(\vec{x})}\right)\mathrm{d}\vec{x} = \int A(\vec{x})L(\vec{x})Q(\vec{x})\mathrm{d}\vec{x}.$$

This integral is approximated by drawing $N_\mathrm{s}$ sample points from the $Q(\vec{x})$ distribution and evaluating

$$\int A(\vec{x})L(\vec{x})Q(\vec{x})\mathrm{d}\vec{x} \approx \frac{1}{N_\mathrm{s}}\sum_{i=1}^{N_\mathrm{s}} A(\vec{x}_i)L(\vec{x}_i), \quad \vec{x}_i \sim Q$$

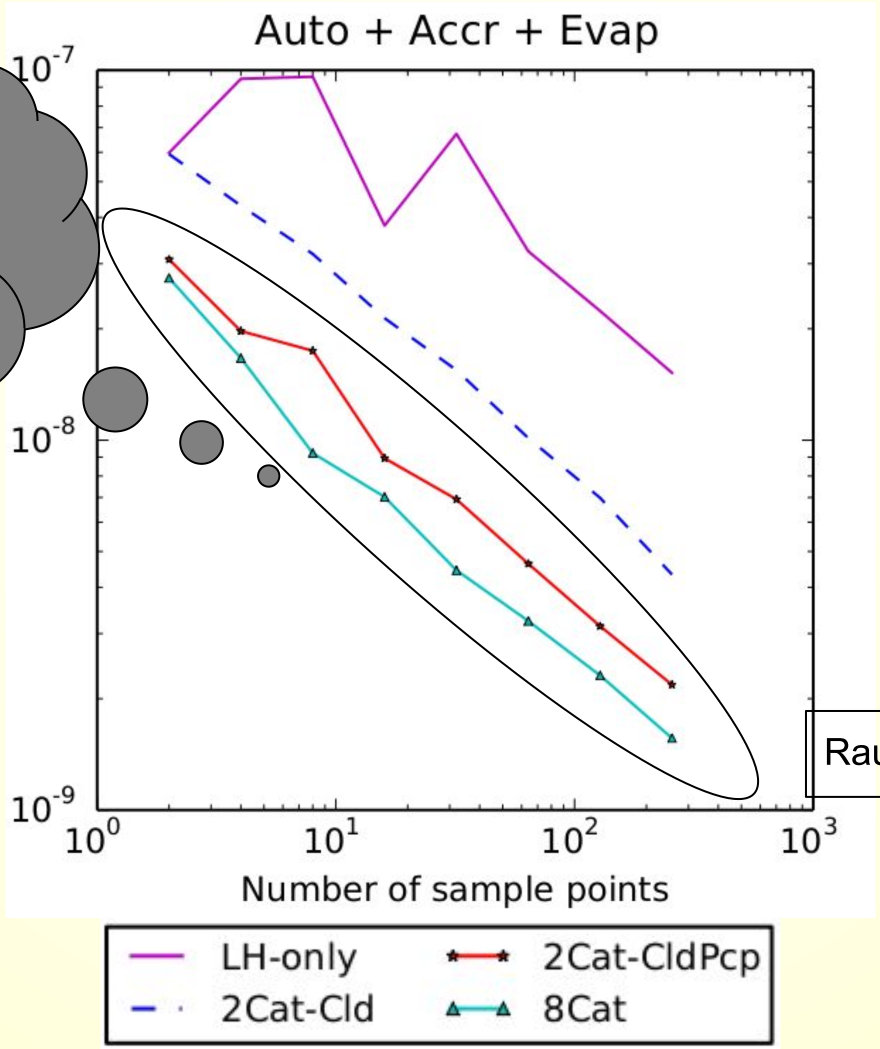
where $\vec{x}_i$ is the $i$th sample point drawn from the $Q(\vec{x})$ distribution.

# Three different strategies for allocating sample points among categories:

- 2Cat-Cld (old method)
  - Allocate 50% of sample points to cloudy regions, and 50% to clear air regions
- 2Cat-CldPcp (new default method)
  - Allocate points ("as many as we can") to regions containing either cloud or precipitation
  - Some points placed in the "boring" region with no precipitation and no cloud to avoid large weights
- 8Cat (new experimental method)
  - All eight category allocations set by user
  - Works well if "optimal" category allocations are similar for many cloud types

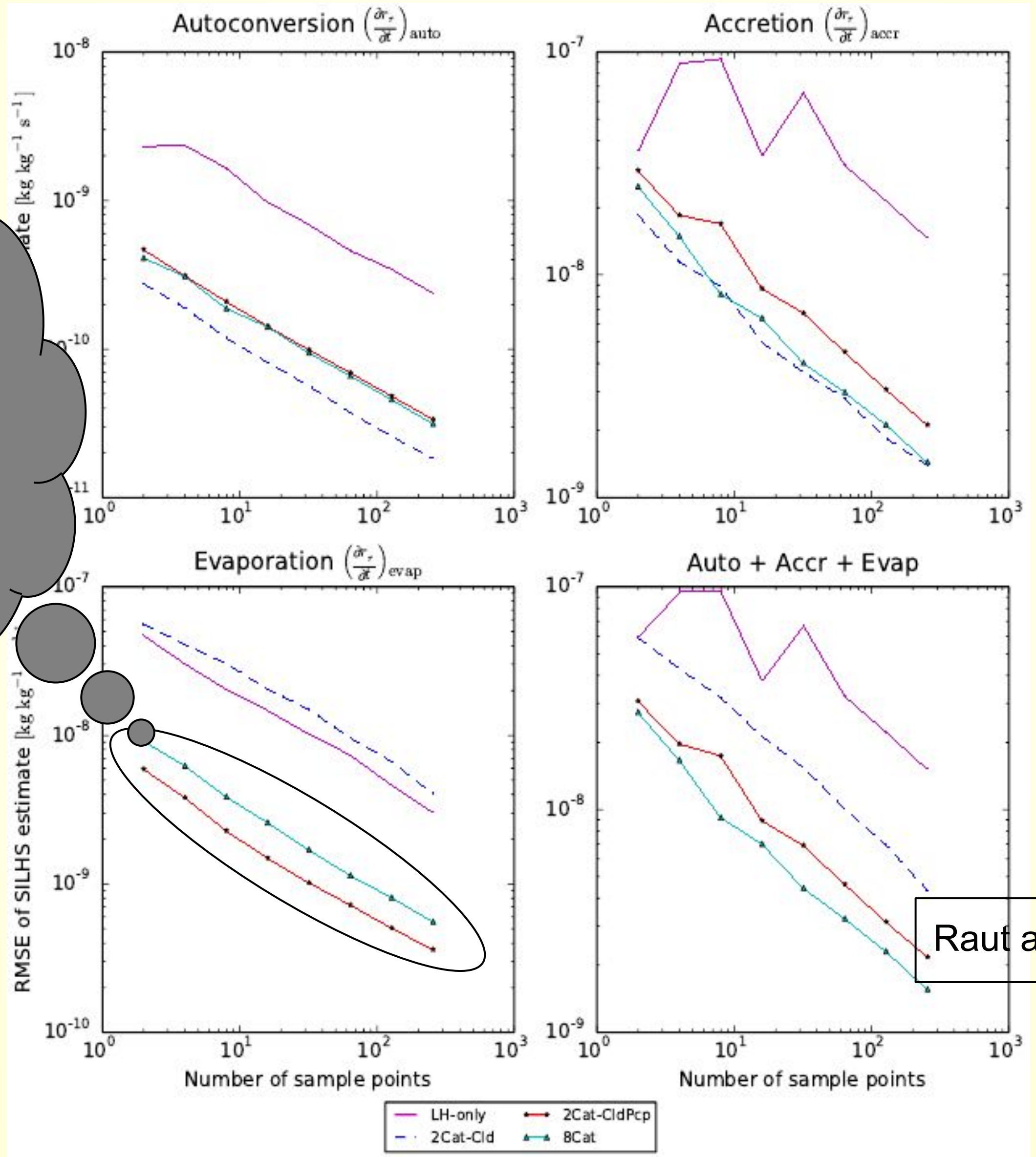
# Single column result (RICO Cu): the new methods improve the rain tendency estimate

Both new methods outperform the old method




Raut and Larson (2016, GMD)

# Key improvement is in estimate of evaporation




RMSE of evaporation is greatly improved (reduced) in the new methods

Raut and Larson (2016, GMD)

# The new method has several advantages

- Flexibility in distributing sample points
  - E.g., the ability to sample out-of-cloud processes preferentially is important for evaporation

- Decreased computational cost
  - Reduced noise: need less points to achieve a desired accuracy in estimation
  - The method itself does not significantly affect computational cost as compared to the old method.

# Conclusions

- We can represent subgrid variability rather generally using a PDF parameterization (CLUBB) plus a subcolumn sampler (SILHS).
- The chief disadvantages of CLUBB-SILHS are its computational expense and noisiness.
- The chief advantages are (i) its ability to handle correlations among variates; (ii) its ability to foster consistency of assumptions about subgrid variability; and (iii) its plug-n-play convenience.

Thanks for your time!