This project AIDA receives funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 776262.

Automated characterization of magnetic reconnection using particle distributions

06 May 2020

Romain Dupuis
KU Leuven

KU LEUVEN

Contributors: R. Dupuis, M. V. Goldman, D. L. Newman, J. Amaya, and G. Lapenta

EGU 2020 – ST 1.9 "Theory and Simulation of Solar System plasmas"

Magnetic reconnection

Magnetic Reconnection

- **Complex physical phenomenon**
  - Break of the frozen-in magnetic field
  - Magnetic energy release
  - Transport mechanism, particle acceleration
  - Magnetic Electron Diffusion Region (EDR)

- **Occurring in many plasma environments**
  - Sun: flares, CME
  - Earth's magnetosphere: magnetopause, magnetotail
  - MMS mission

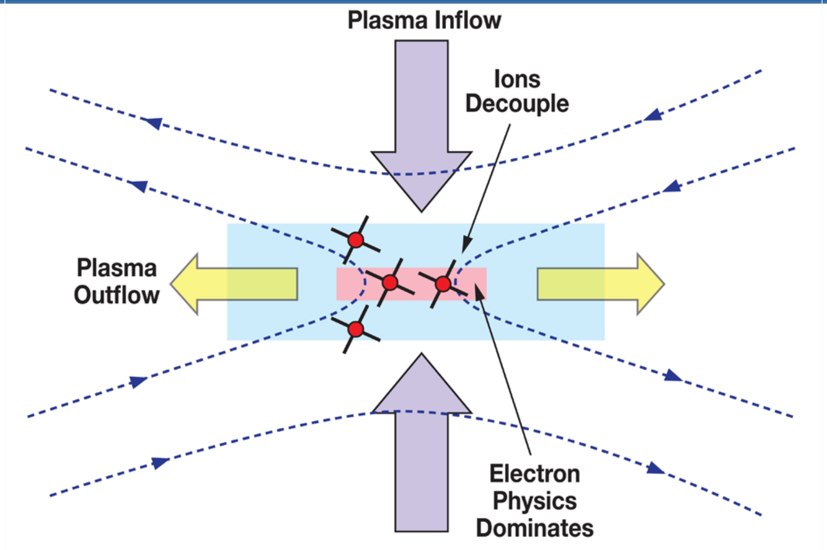
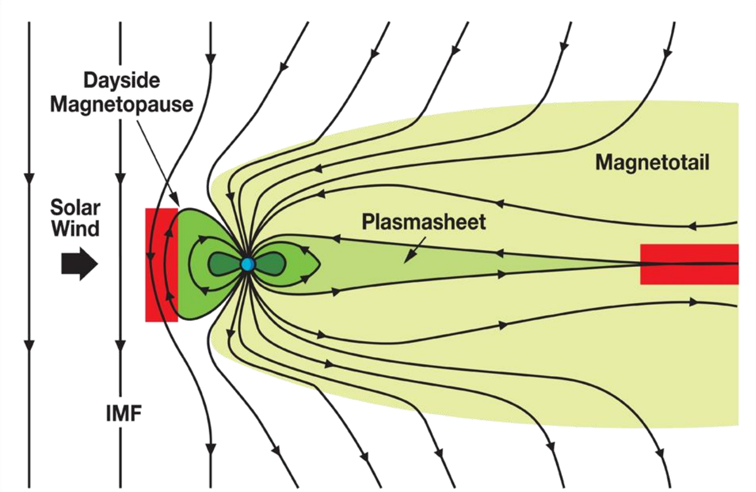


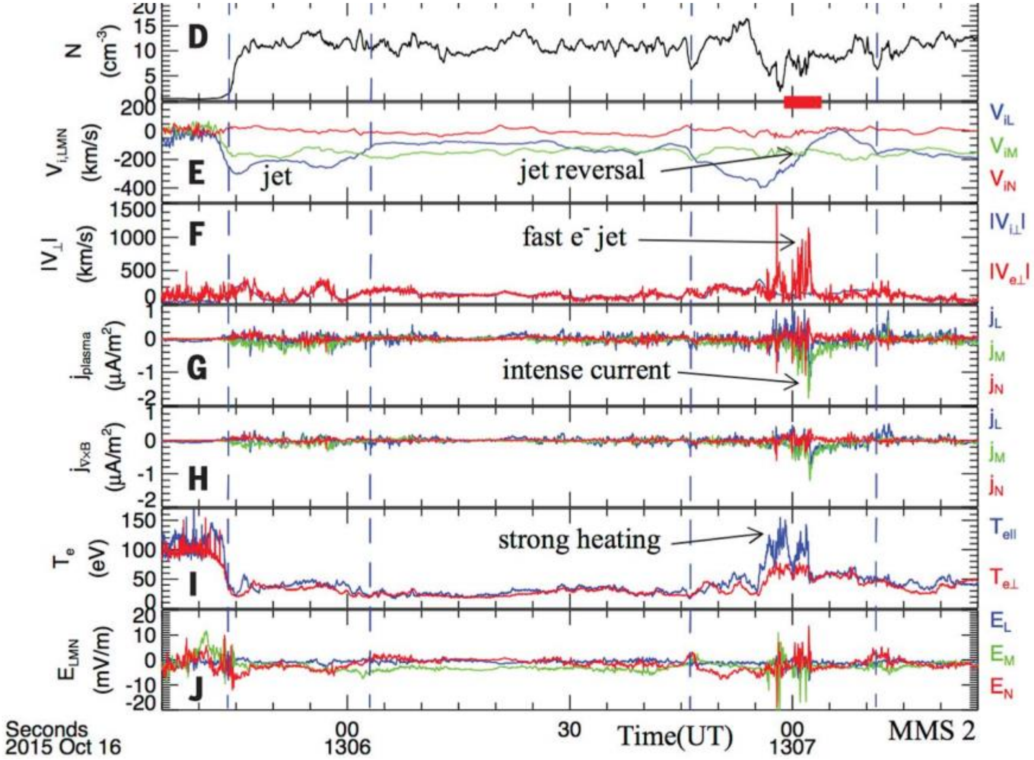
Image Credit: NASA MMS



Detection of reconnection

- **EDR is very small**
  - The precise detection is hard
  - Use of indirect signatures

- **Usually two groups**
  - Field quantities
  - Statistical moments

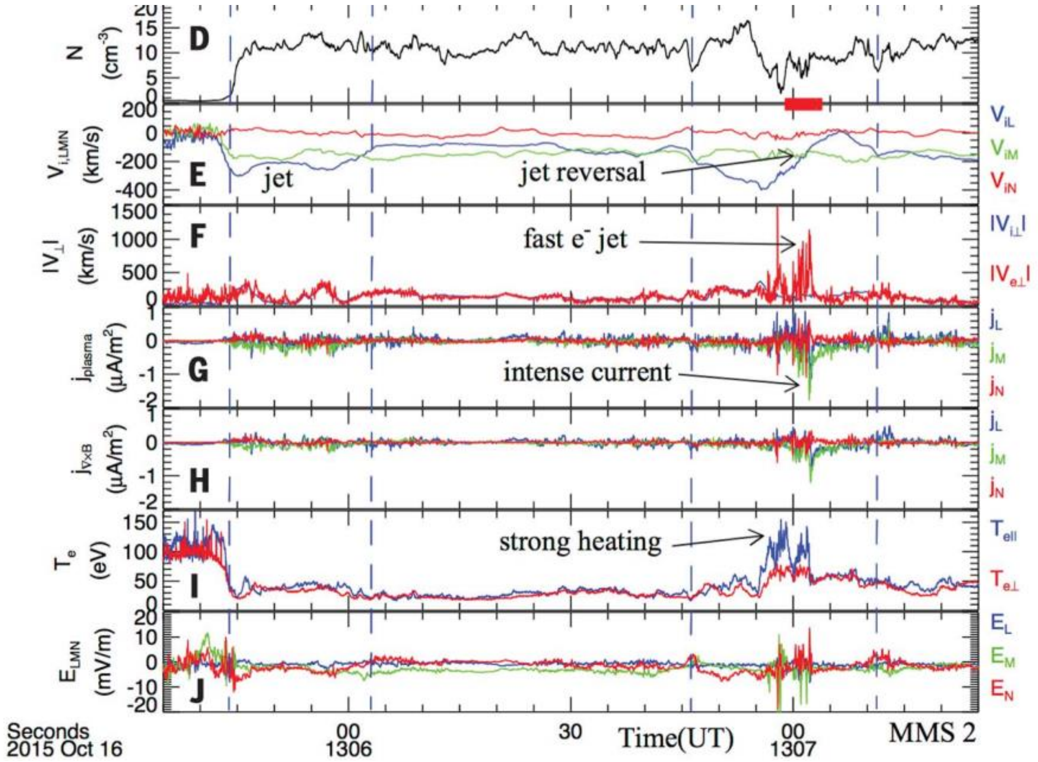


Burch et al. 2016

Detection of reconnection

- **EDR is very small**
  - The precise detection is hard
  - Use of indirect signatures

- **Usually two groups**
  - Field quantities
  - Statistical moments



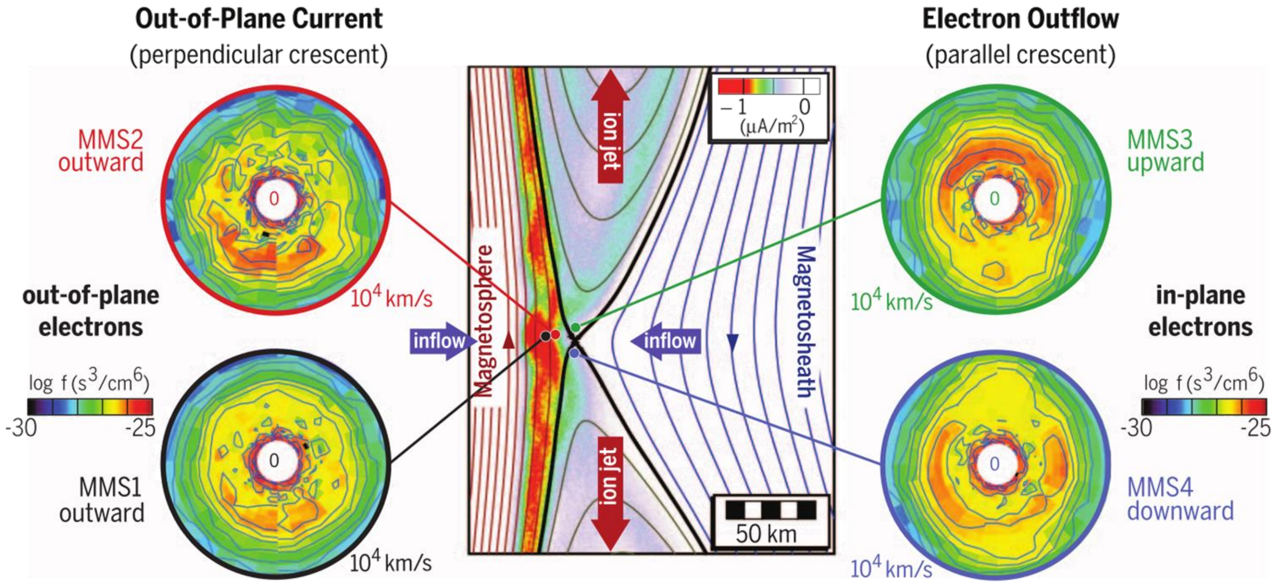
Burch et al. 2016

Third approach: using directly the distribution

Particle distributions

- **Rich part of the information**
  - MMS mission: crescent shape
  - Electron dynamics dominates
  - Beams, power law, top-hat

- **But very large data**
  - 3D velocity space
  - Spatial space
  - Temporal aspect

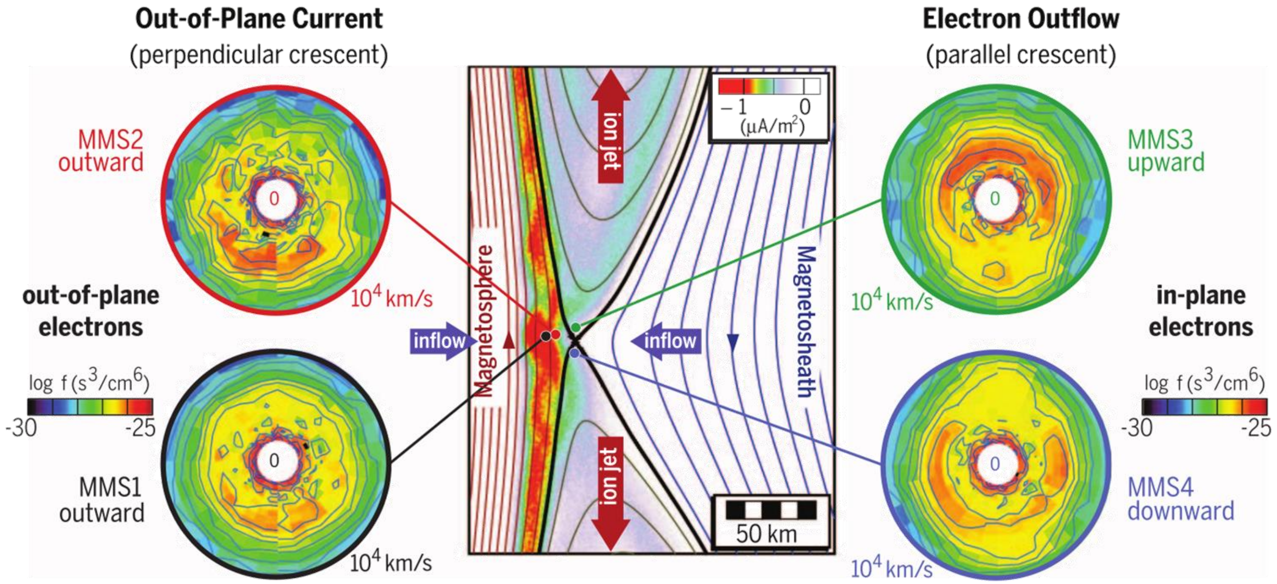


Burch et al. 2016

Particle distributions

- **Rich part of the information**
  - MMS mission: crescent shape
  - Electron dynamics dominates
  - Beams, power law, top-hat

- **But very large data**
  - 3D velocity space
  - Spatial space
  - Temporal aspect



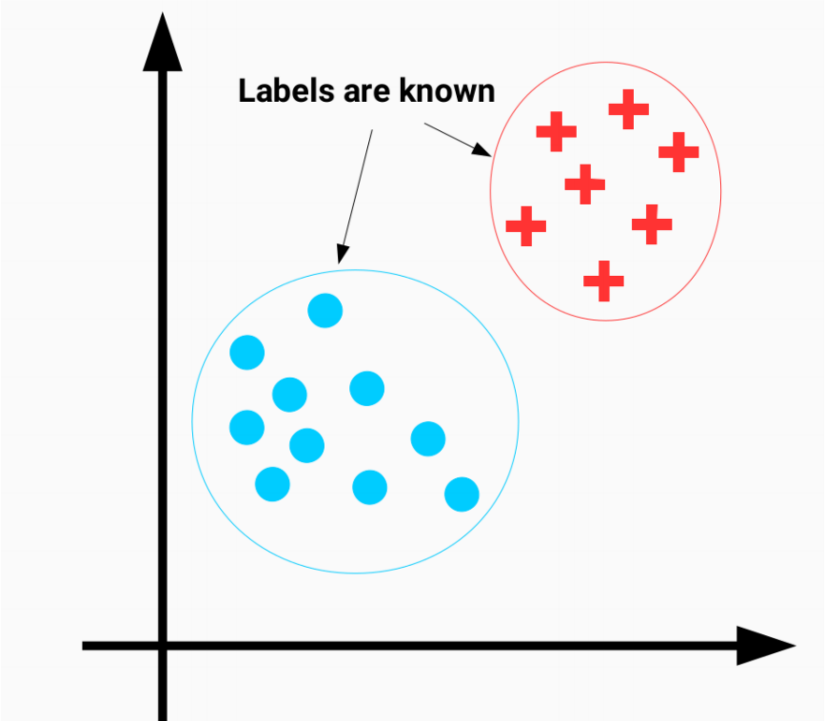
Burch et al. 2016

Extract automatically information from the particle distributions

Machine learning

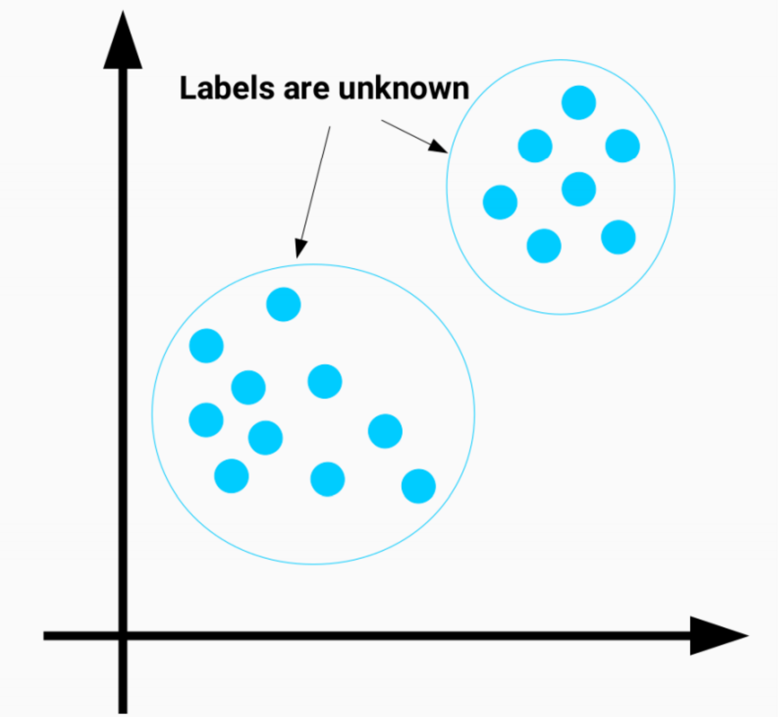
Supervised vs. unsupervised

- **Supervised learning**
  - Regression
  - Classification



Credit S. Carrazza

- **Unsupervised learning**
  - Clustering
  - Dimension reduction
  - Density estimation



Credit S. Carrazza

Related work

- *Automated classification of plasma regions using 3D particle energy distribution. MMS mission: crescent shape,* Olshevsky V. et al, 2019

- *Automatic Detection of Magnetospheric Regions around Saturn using Cassini Data, Yeakel, K et al., 2017*

- *Automatic detection of magnetopause reconnection diffusion regions, Garnier P. et al.*

Table 2. Training Dataset

Label	Region	# examples	percentage
−1	Unknown	0	0
0	Solar Wind	12,135	42.4
1	Foreshock	7,351	25.7
2	Magnetosheath	7,009	24.5
3	Magnetosphere	2,146	7.5
	Total	28,641	100

We want to privilege unsupervised approaches

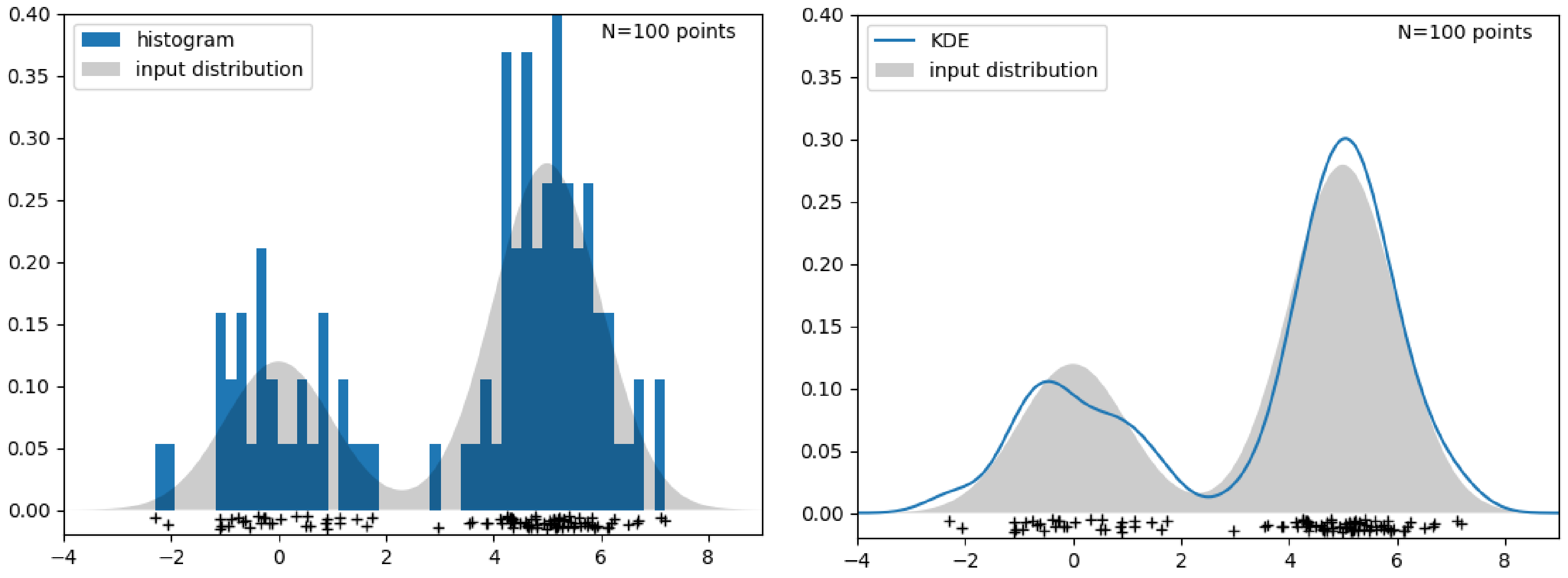
Density estimation

- **Building an estimate of the probability density function**

Density estimation

- **Building an estimate of the probability density function**

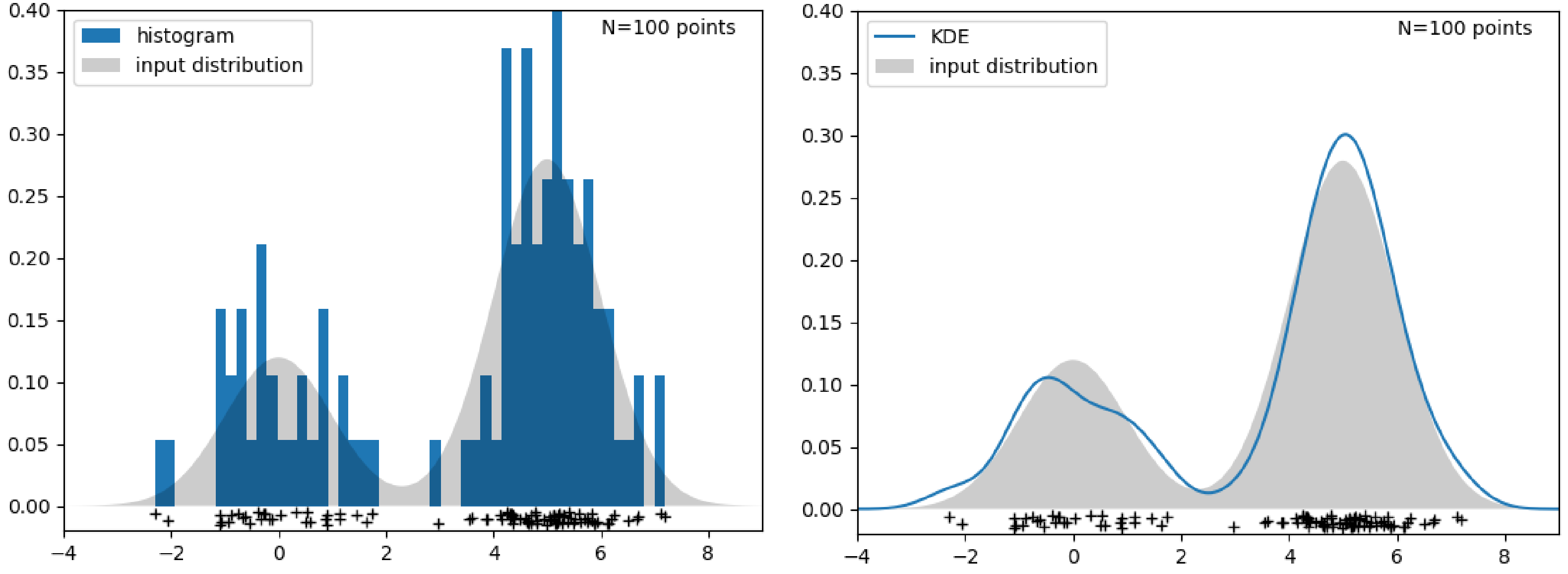
- **Non-parametric methods**
  - Histogram
  - Kernel Density Estimation



Density estimation

- **Building an estimate of the probability density function**

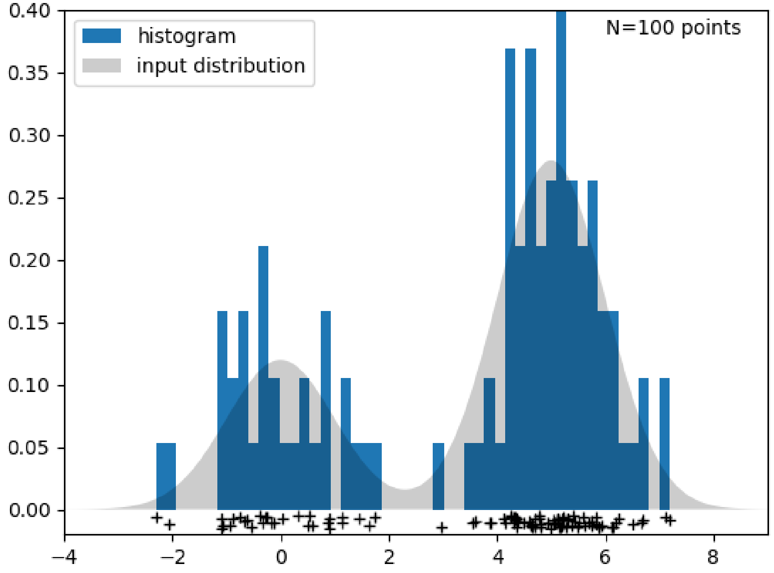
- **Non-parametric methods**
  - Histogram
  - Kernel Density Estimation

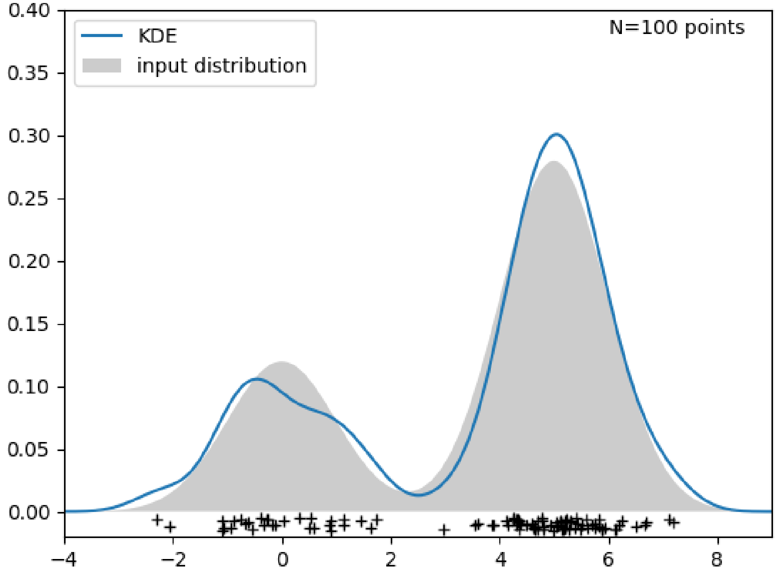


- **Parametric methods**
  - Fitting given distributions
  - Gaussian Mixture Models (GMM)

Density estimation

- **Building an estimate of the probability density function**





- **Non-parametric methods**
  - Histogram
  - Kernel Density Estimation

- **Parametric methods**
  - Fitting given distributions
  - Gaussian Mixture Models (GMM)

Gaussian Mixture models

- **Gaussian probability distribution**

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right)$$

Gaussian Mixture models

- **Gaussian probability distribution**

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\,\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right)$$

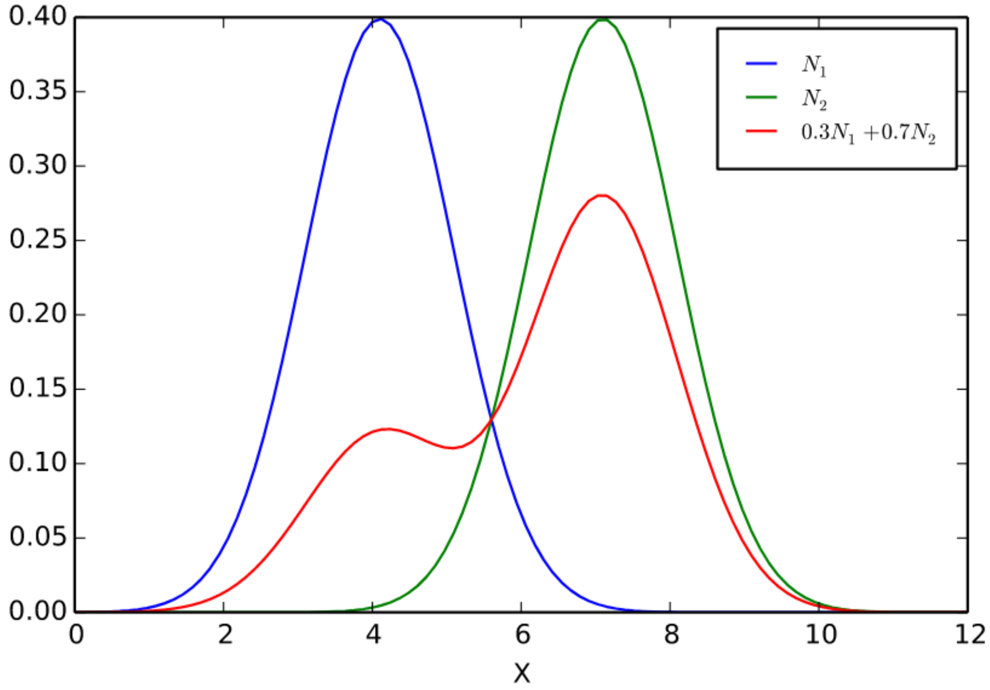
- **Sum of Gaussians (mixture)**

$$p(\boldsymbol{x}|\boldsymbol{\Phi}) = \sum_{k=1}^{K} w_k\,\mathcal{N}(\boldsymbol{x}|\boldsymbol{\theta_k})$$

- **Parameters to find**

$$\boldsymbol{\theta_k} = (w_k\,,\boldsymbol{\mu_k}\,,\boldsymbol{\Sigma_k})$$

$p(x) = 0.3N_1 + 0.7N_2$



Credit Rémi Emonet

Gaussian Mixture models

- **Gaussian probability distribution**

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\,\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right)$$

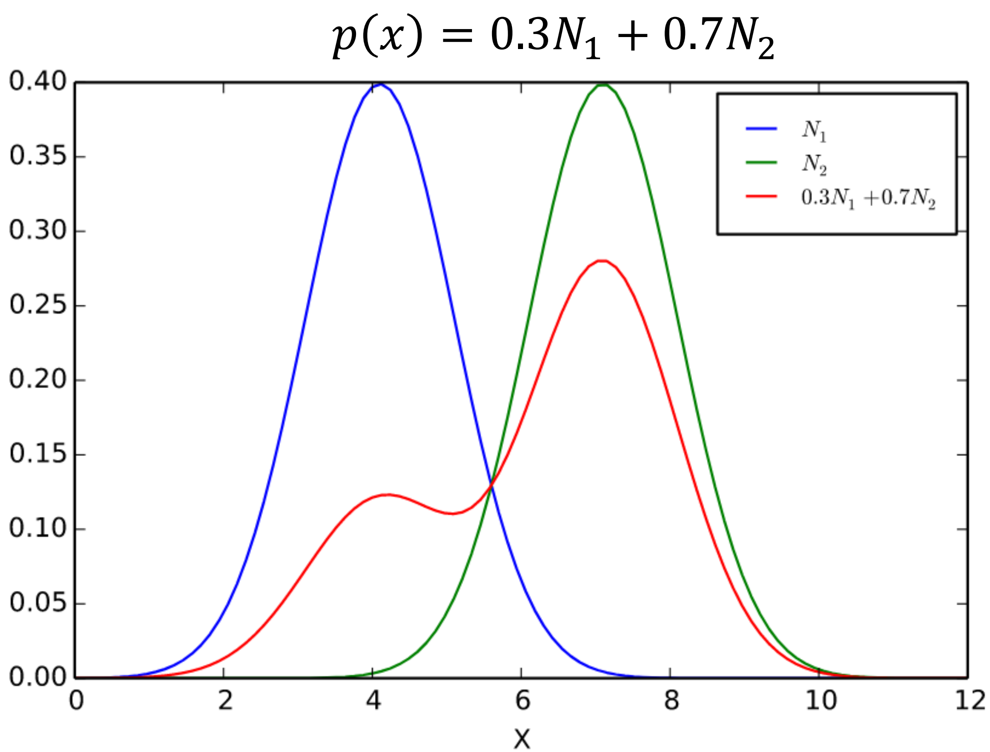
- **Sum of Gaussians (mixture)**

Number of components

$$p(\boldsymbol{x}|\boldsymbol{\Phi}) = \sum_{k=1}^{K} w_k\,\mathcal{N}(\boldsymbol{x}|\boldsymbol{\theta_k})$$

- **Parameters to find**

$$\boldsymbol{\theta_k} = (w_k,\boldsymbol{\mu_k},\boldsymbol{\Sigma_k})$$



Credit Rémi Emonet

Gaussian Mixture Model

$$l(\phi;\boldsymbol{X}) = \sum_{i=1}^{n} \ln f(\boldsymbol{X_i}, \phi)$$

- **How to infer the best parameters ? Maximum likelihood estimation**

$$l(\phi;\boldsymbol{X}) = \sum_{i=1}^{n} \ln \left[\sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x_i}|\boldsymbol{\theta_k}) \right]$$

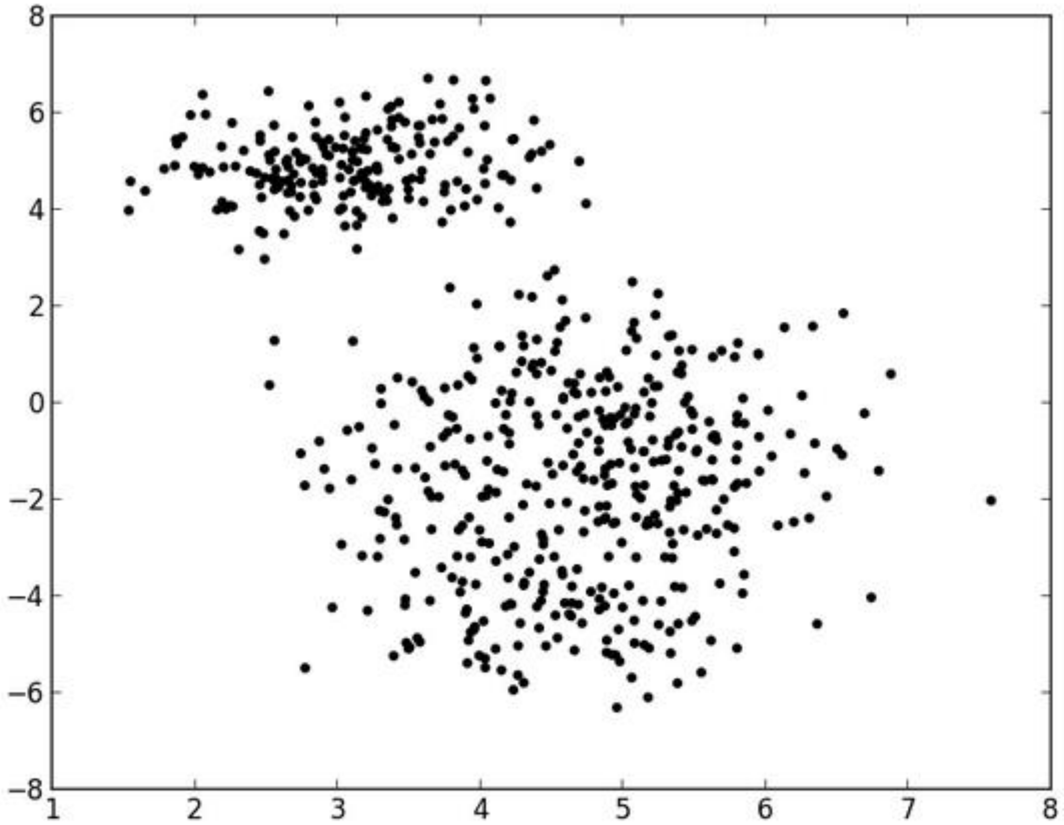
- **Maximizing the likelihood estimation**
  - Non linear maximization problem
  - No closed form

$$\frac{\partial l}{\partial \boldsymbol{\mu_k}} = 0 \qquad \frac{\partial l}{\partial \boldsymbol{\Sigma_k}} = 0 \qquad \frac{\partial l}{\partial w_k} = 0 \; s.t. \sum_{k=1}^{K} w_k = 1$$

- **Need to find numerical local maximum: Expectation Maximization**

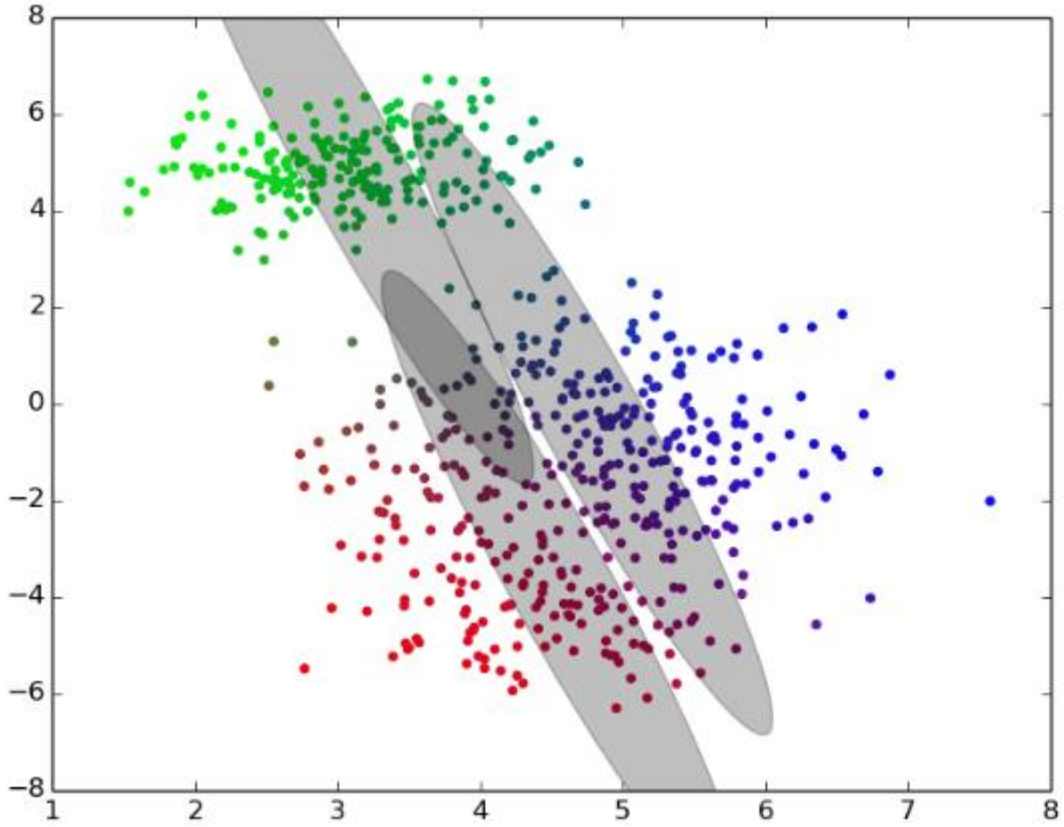
Expectation Maximization

- **Very effective for models with unobserved latent variables**



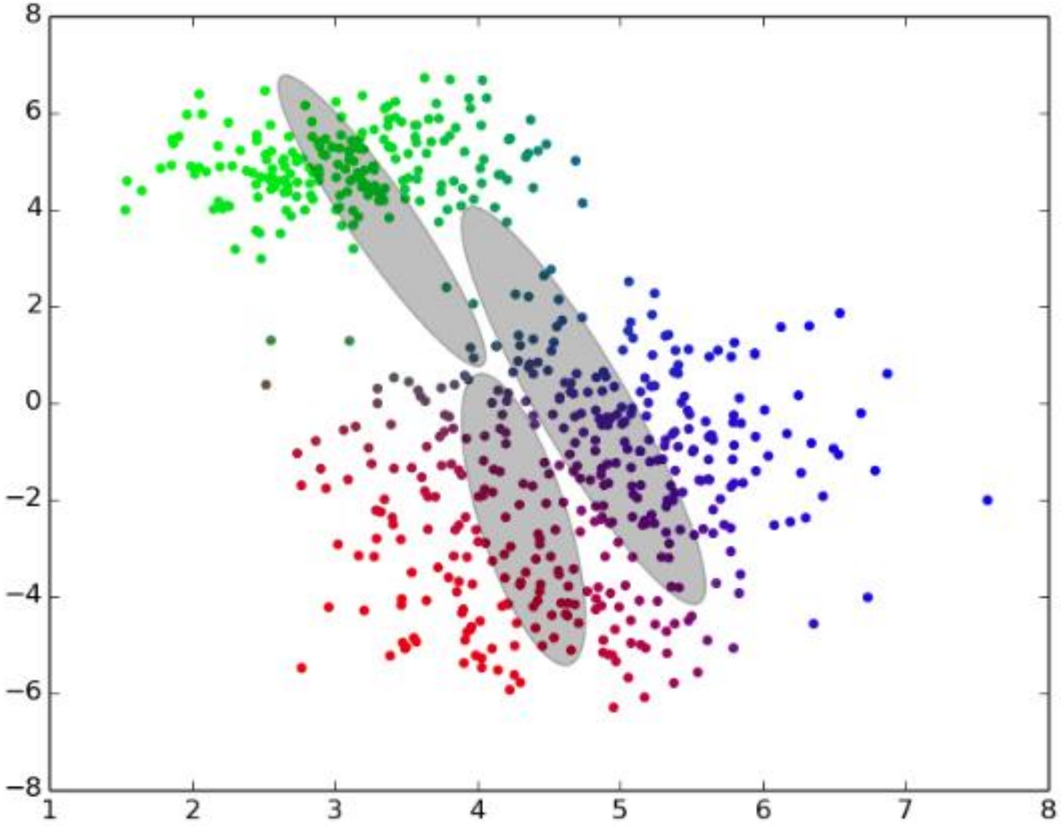
Expectation Maximization

- **Very effective for models with unobserved latent variables**



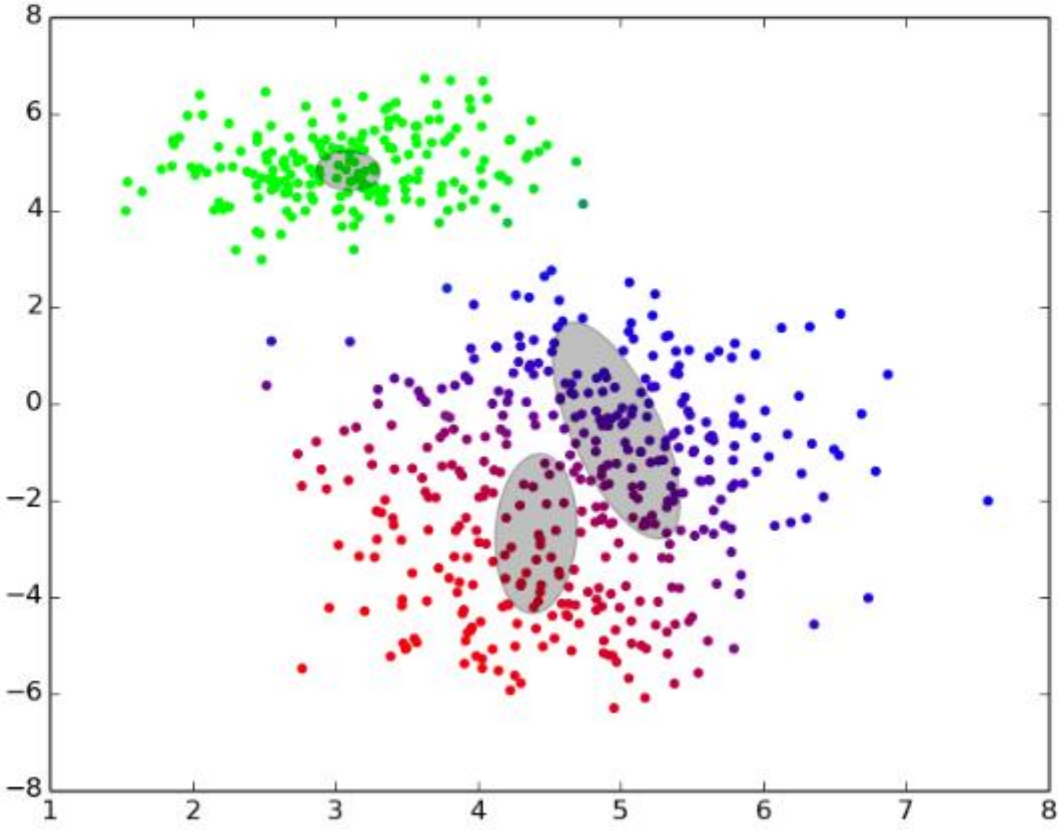
Expectation Maximization

- **Very effective for models with unobserved latent variables**



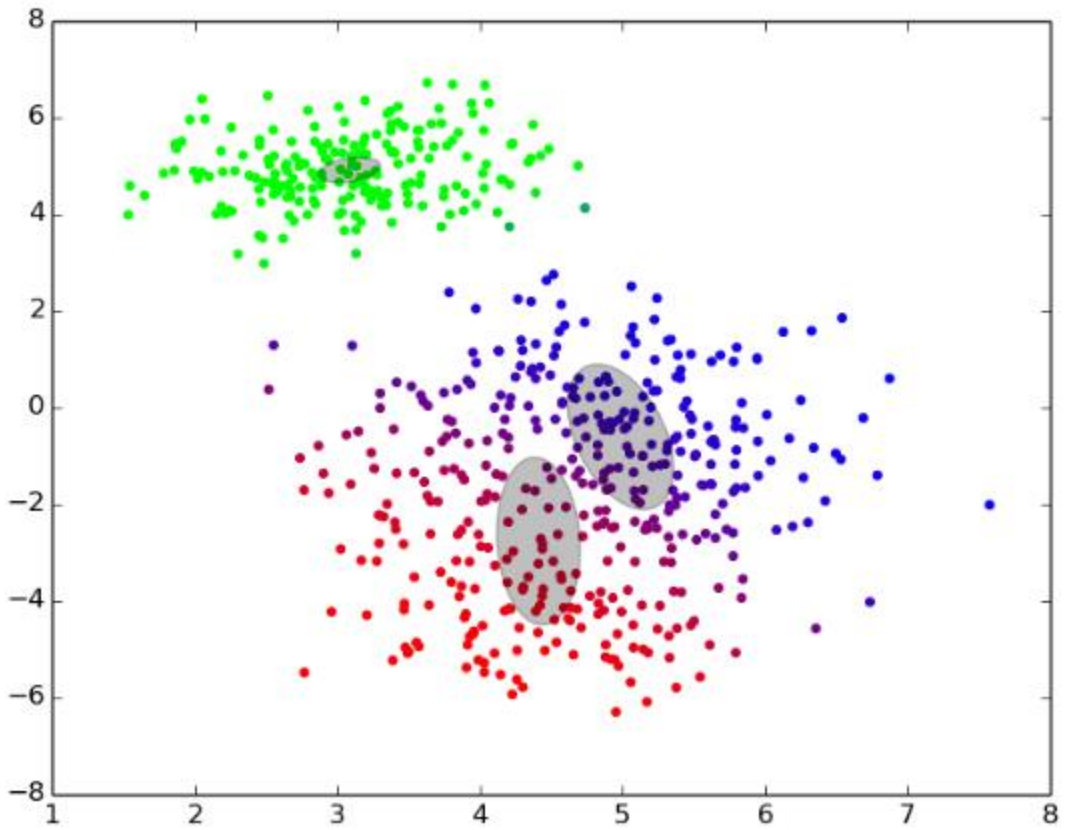
Expectation Maximization

- **Very effective for models with unobserved latent variables**



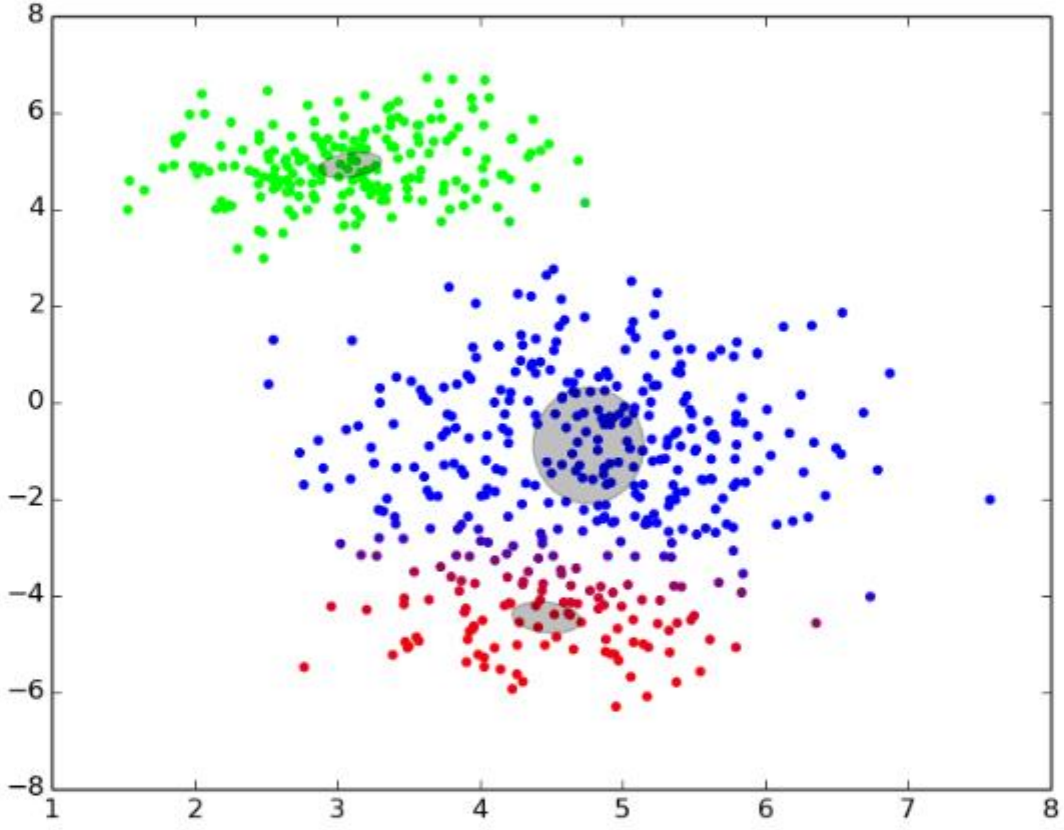
Expectation Maximization

- **Very effective for models with unobserved latent variables**



Expectation Maximization

- **Very effective for models with unobserved latent variables**



Model selection

- **How to determine the number of components K?**

Model selection

- **How to determine the number of components K?**

- **Information theory**
  - Aikaike Information Criterion
  - Bayesian Information Criterion

$$AIC = 2k - 2\ln(L)$$
$$BIC = \ln(n)k - 2\ln(L)$$

Model selection

- **How to determine the number of components K?**

- **Information theory**
  - Aikaike Information Criterion
  - Bayesian Information Criterion

$$AIC = 2k - 2\ln(L)$$
$$BIC = \ln(n)k - 2\ln(L)$$

Complexity

Goodness of fit

Model selection

- **How to determine the number of components K?**

- **Information theory**
  - Aikaike Information Criterion
  - Bayesian Information Criterion

$$AIC = 2k - 2\ln(L)$$
$$BIC = \ln(n)k - 2\ln(L)$$

Complexity

Goodness of fit

- **Various interpretation to the number of components K**
  - Beams/electron subpopulation
  - Complex distribution
  - Deviation from a Gaussian (tail, mode width, etc.)

Simulations

Simulations

- **Access to the complete description of the plasma over all the spatial grid**

- **2.5D collisionless Particle In Cell simulations**
  - iPic3D (Markidis et al.)
  - Double Harris sheet case, weak guide field
  - Grid: 769 x 1025 (30di x 40 di)
  - 196,000,000 particles (~250 particles/cell)
  - Weighted particle injection

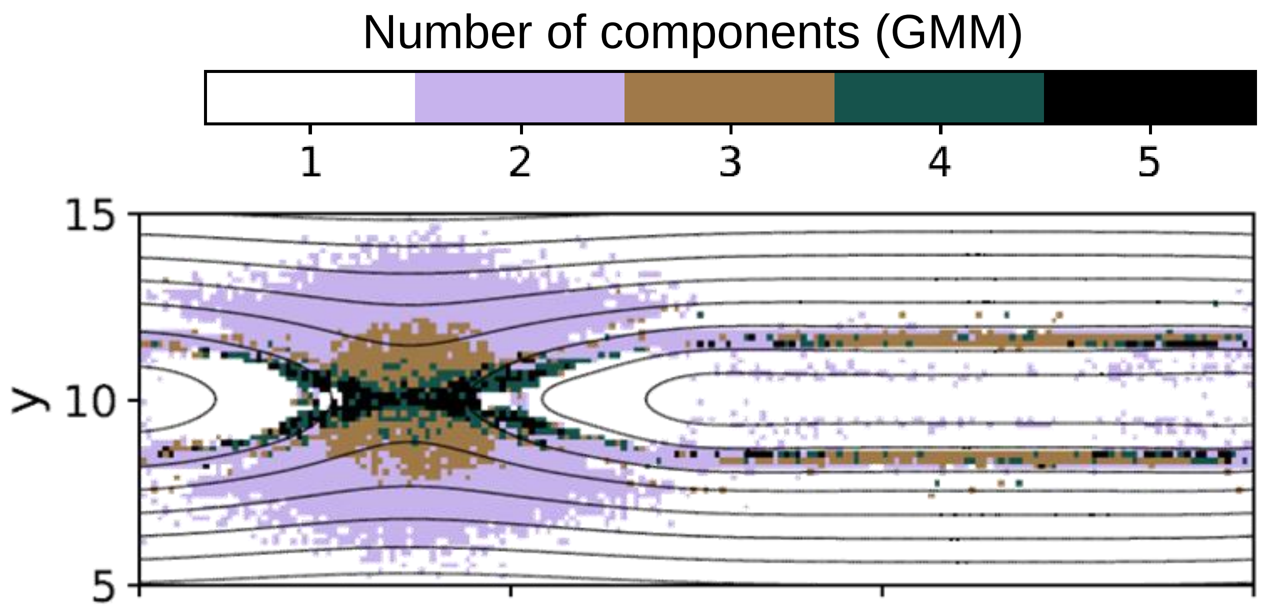
- **B-field-aligned basis**

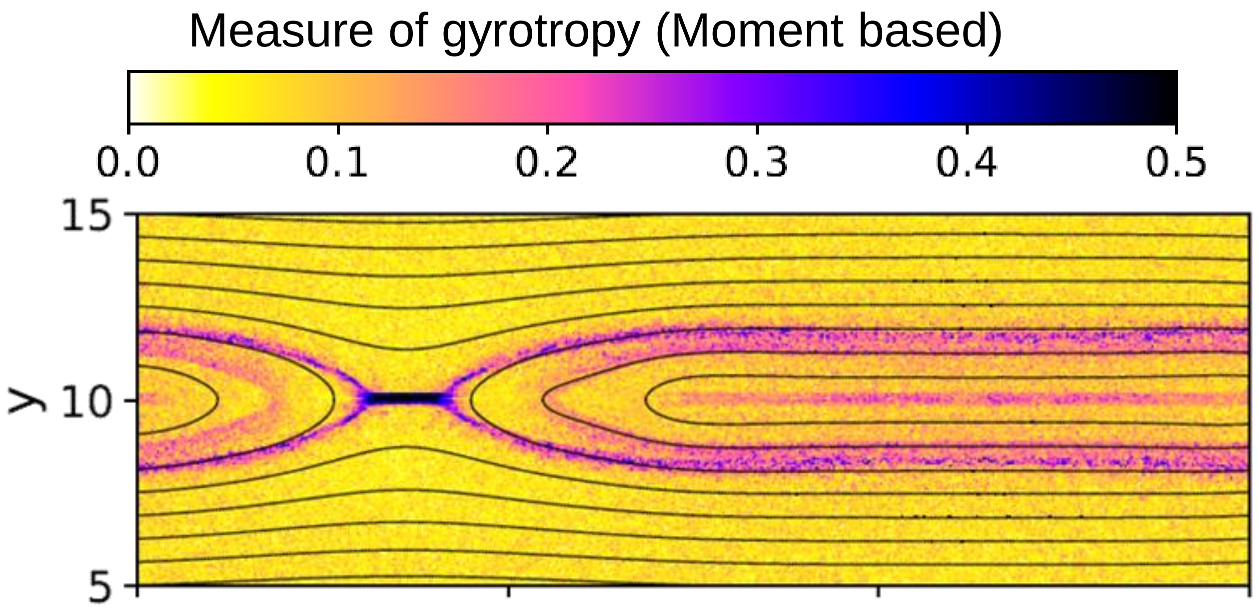
$$e_{\|} := \widehat{\boldsymbol{B}}, \text{ where } \widehat{\boldsymbol{B}} = \frac{\boldsymbol{B}}{\|\boldsymbol{B}\|}$$

$$\boldsymbol{e}_{\perp 1} := \widehat{\boldsymbol{B}} \times \boldsymbol{e_z}$$

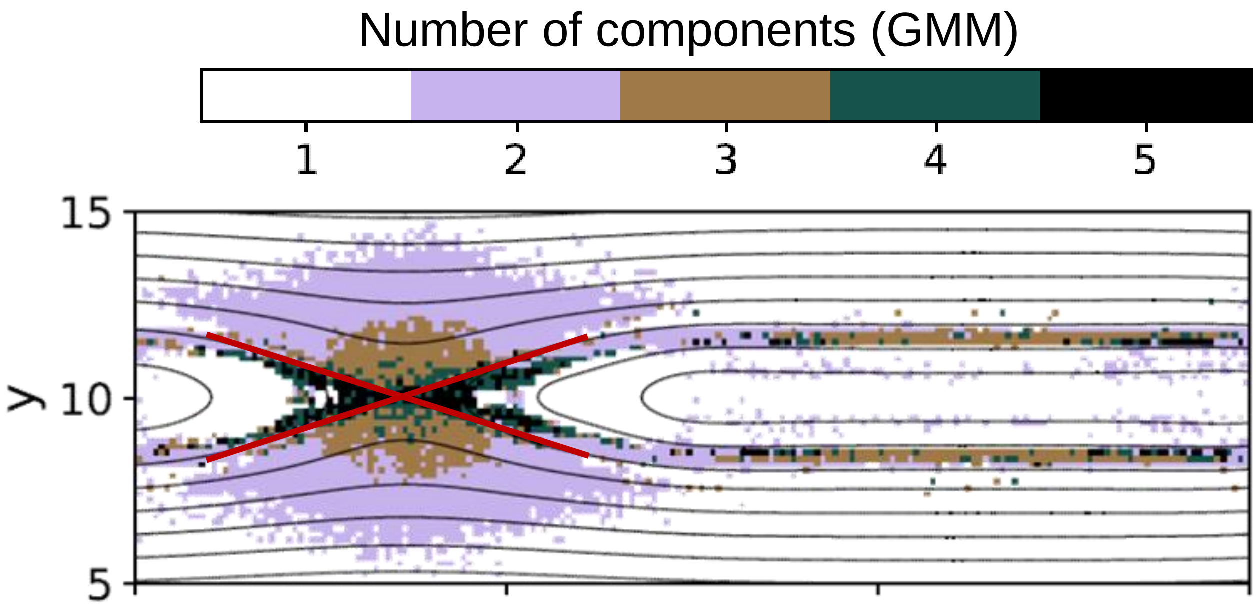
$$\boldsymbol{e}_{\perp 2} := \widehat{\boldsymbol{B}} \times \boldsymbol{e}_{\perp 1} = -\widehat{B}^2 \boldsymbol{e_z} + (\boldsymbol{e_z}.\widehat{\boldsymbol{B}})\widehat{\boldsymbol{B}}$$

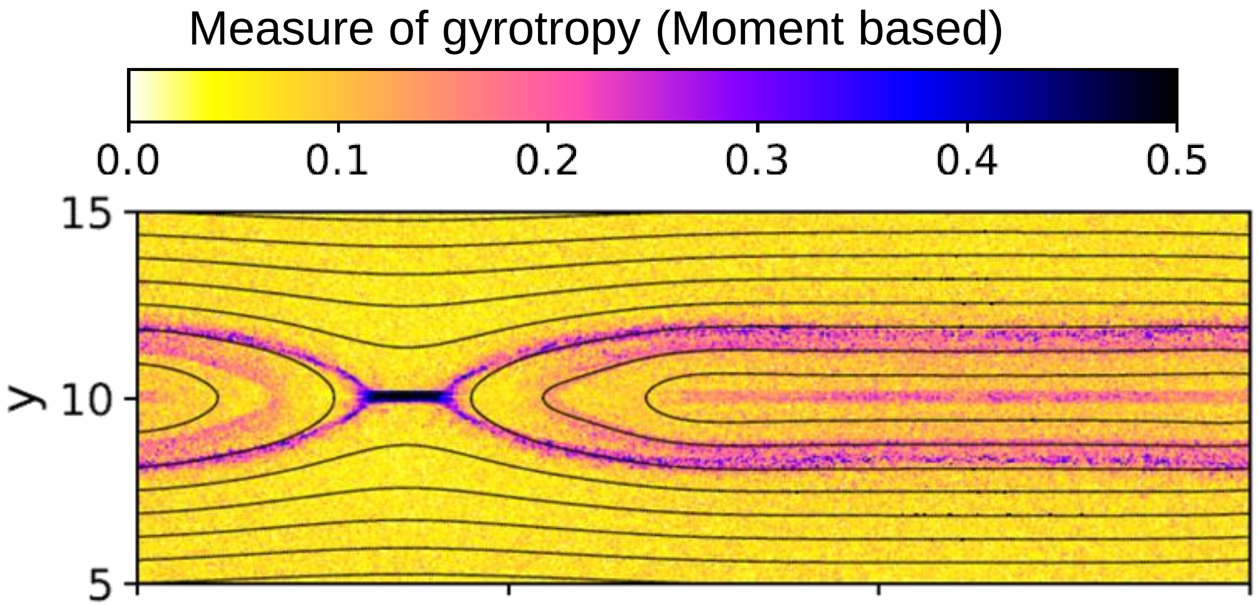
Number of components



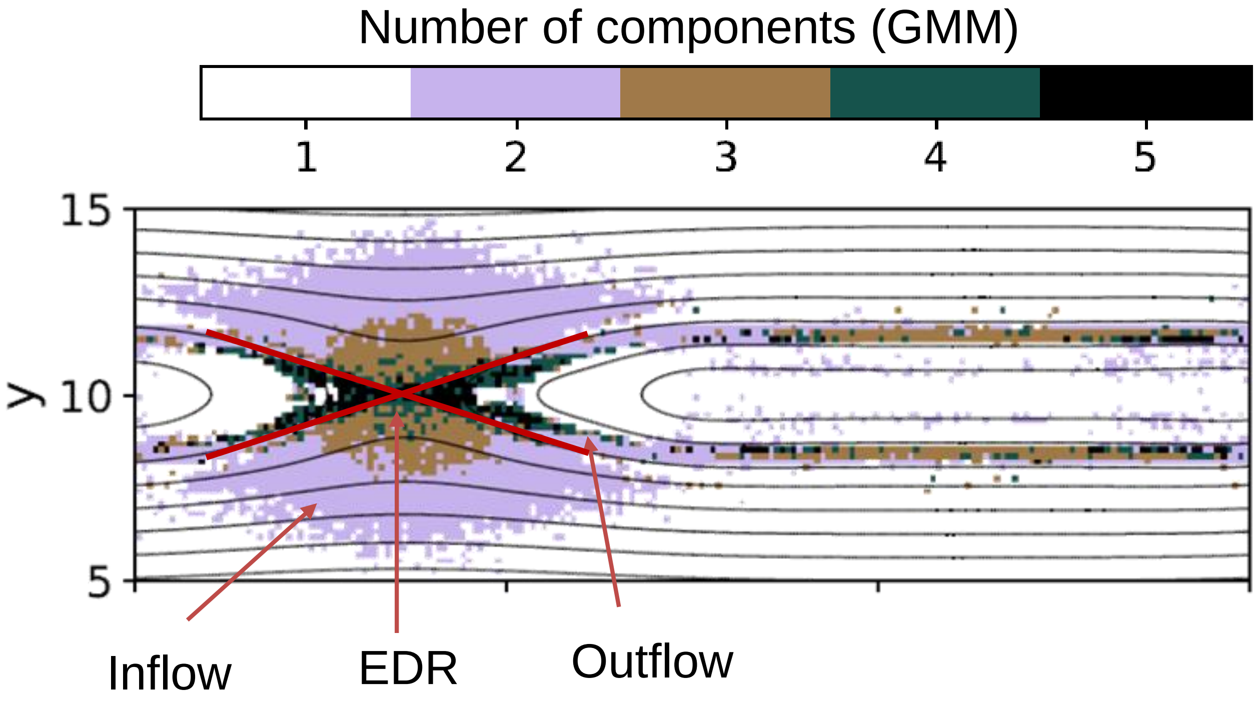


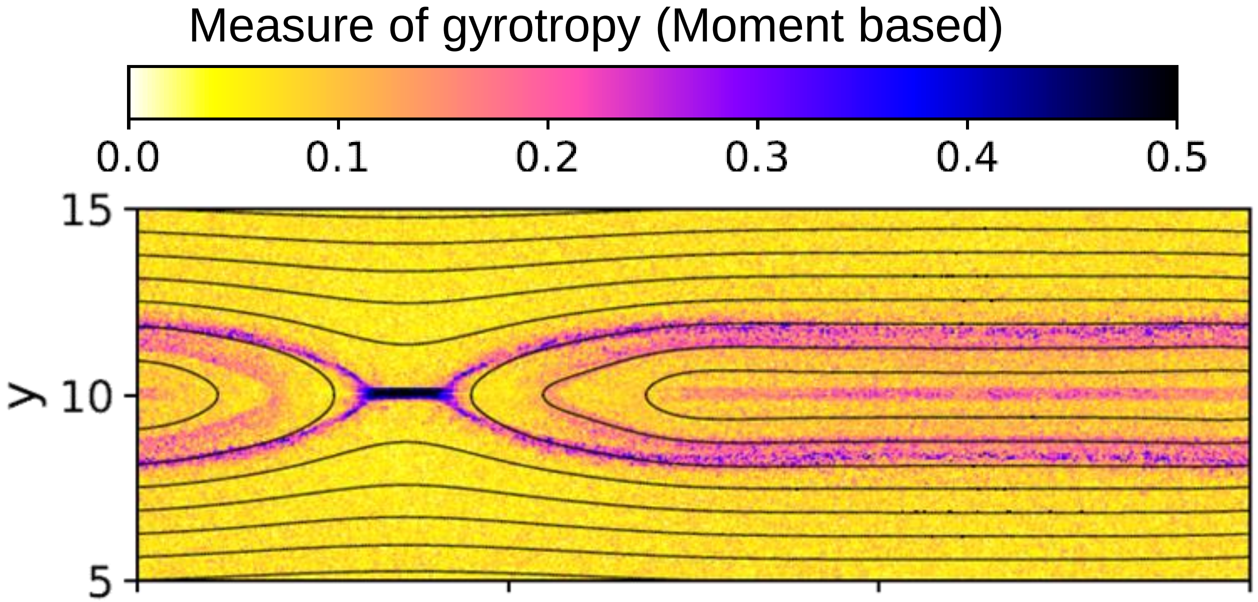
Number of components



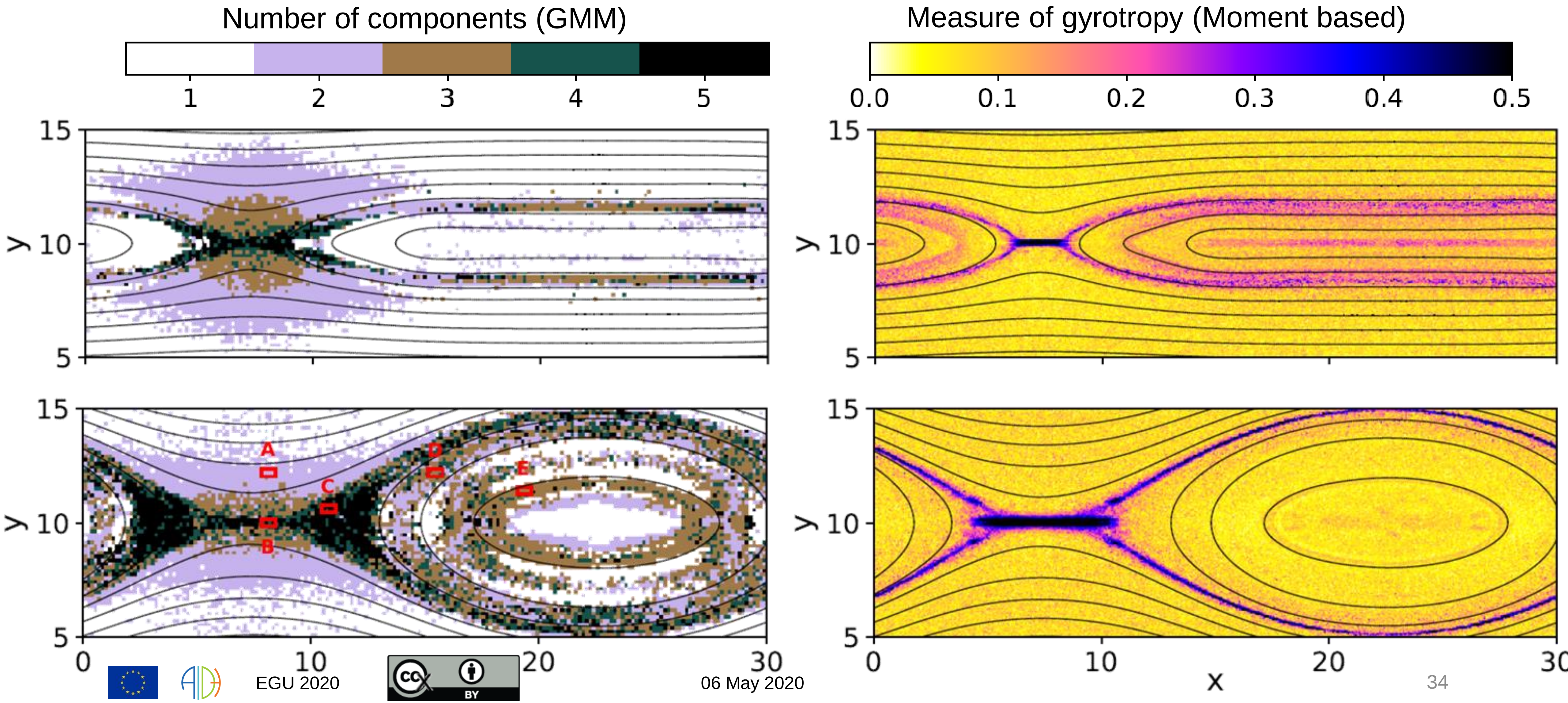


Number of components





Number of components



Analyzing the mixtures

- **Thermal energy of the distribution**

$$E_{thermal} = \frac{1}{N_p} \sum_{i=1}^{3} \left[\sum_p \left(\boldsymbol{V_p} - \langle \boldsymbol{V_p} \rangle \right)^2 \right]_i , \text{ with } \langle \boldsymbol{V_p} \rangle = \sum_p \frac{\boldsymbol{V_p}}{N_p}$$

Analyzing the mixtures

- **Thermal energy of the distribution**

$$E_{thermal} = \frac{1}{N_p} \sum_{i=1}^{3} \left[\sum_{p} \left(\boldsymbol{V_p} - \langle \boldsymbol{V_p} \rangle \right)^2 \right]_i, \text{ with } \langle \boldsymbol{V_p} \rangle = \sum_{p} \frac{\boldsymbol{V_p}}{N_p}$$

- **Variance of the mixture:**

$$(\sigma^2)^{(K)} = \sum_{i=1}^{3} \left[\sum_{k=1}^{K} w_k^2 \left(\boldsymbol{\sigma_k} \right)^2 + \sum_{k=1}^{K} w_k \left(\boldsymbol{\mu_k} \right)^2 - \left(\sum_{k=1}^{K} w_k (\boldsymbol{\mu_k}) \right)^2 \right]_i$$

Analyzing the mixtures

- **Thermal energy of the distribution**

$$E_{thermal} = \frac{1}{N_p} \sum_{i=1}^{3} \left[\sum_p \left(\boldsymbol{V_p} - \langle \boldsymbol{V_p} \rangle \right)^2 \right]_i , \text{ with } \langle \boldsymbol{V_p} \rangle = \sum_p \frac{\boldsymbol{V_p}}{N_p}$$

- **Variance of the mixture:**

$$(\sigma^2)^{(K)} = \sum_{i=1}^{3} \left[\sum_{k=1}^{K} w_k^2 \left(\boldsymbol{\sigma_k} \right)^2 + \sum_{k=1}^{K} w_k \left(\boldsymbol{\mu_k} \right)^2 - \left(\sum_{k=1}^{K} w_k (\boldsymbol{\mu_k}) \right)^2 \right]_i$$

Analyzing the mixtures: energy drop and deviation

- **Thermal energy of the distribution**

$$E_{thermal} = \frac{1}{N_p} \sum_{i=1}^{3} \left[\sum_p \left(\boldsymbol{V_p} - \langle \boldsymbol{V_p} \rangle \right)^2 \right]_i , \text{ with } \langle \boldsymbol{V_p} \rangle = \sum_p \frac{\boldsymbol{V_p}}{N_p}$$

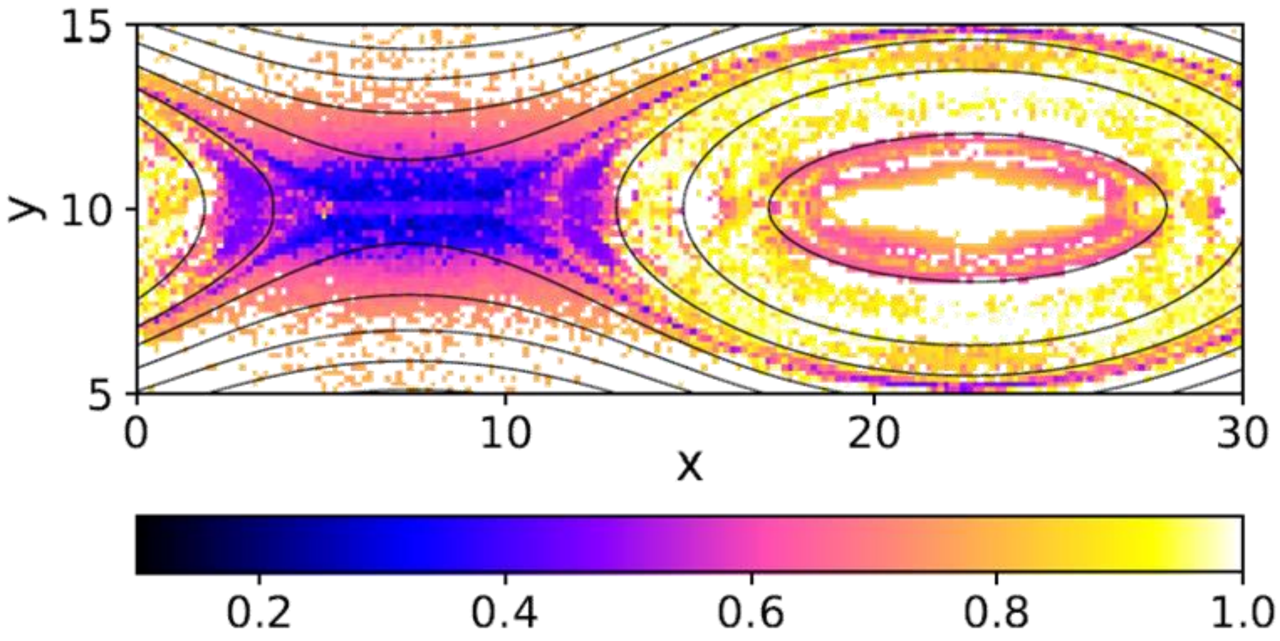
- **Variance of the mixture:**

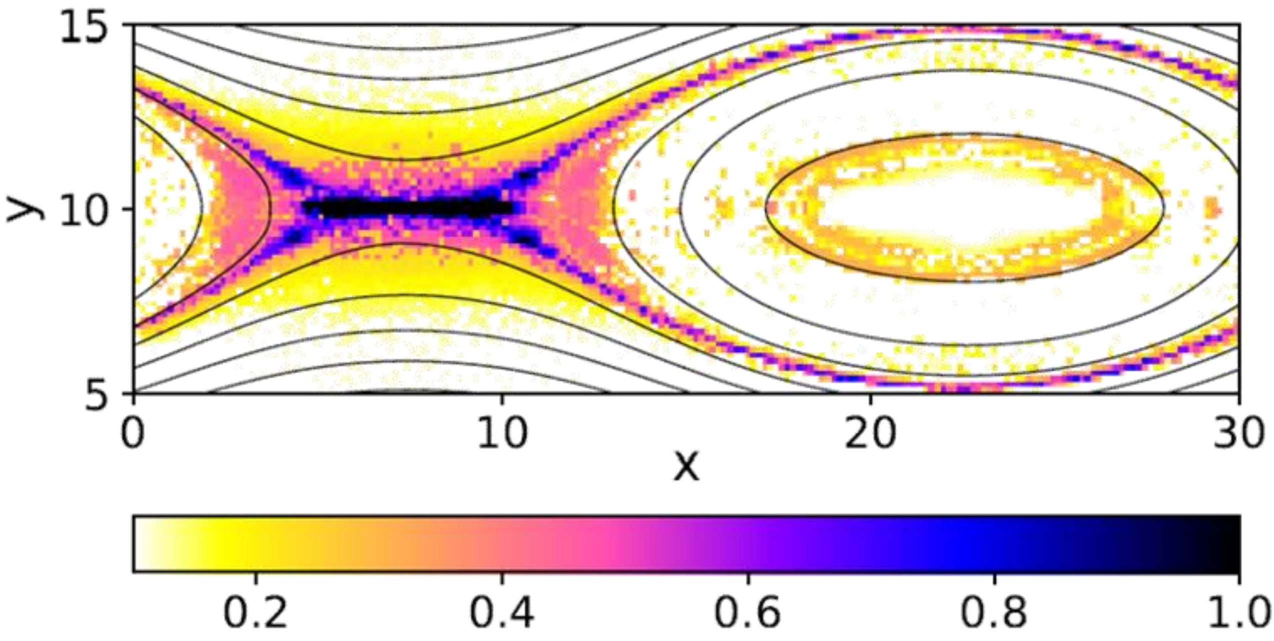
$$(\sigma^2)^{(K)} = \sum_{i=1}^{3} \left[\underbrace{\sum_{k=1}^{K} w_k^2 \left(\boldsymbol{\sigma_k} \right)^2}_{E^{(K)}_{thermal}} + \underbrace{\sum_{k=1}^{K} w_k \left(\boldsymbol{\mu_k} \right)^2 - \left(\sum_{k=1}^{K} w_k (\boldsymbol{\mu_k}) \right)^2}_{E^{(K)}_{dev}} \right]_i$$

- **Diagnostic quantities**

$$E_{drop} = \frac{E^{(K)}_{thermal}}{E_{thermal}} \qquad E_{dev} = \frac{E^{(K)}_{dev}}{E_{thermal}}$$

Analyzing the mixtures: energy drop and deviation





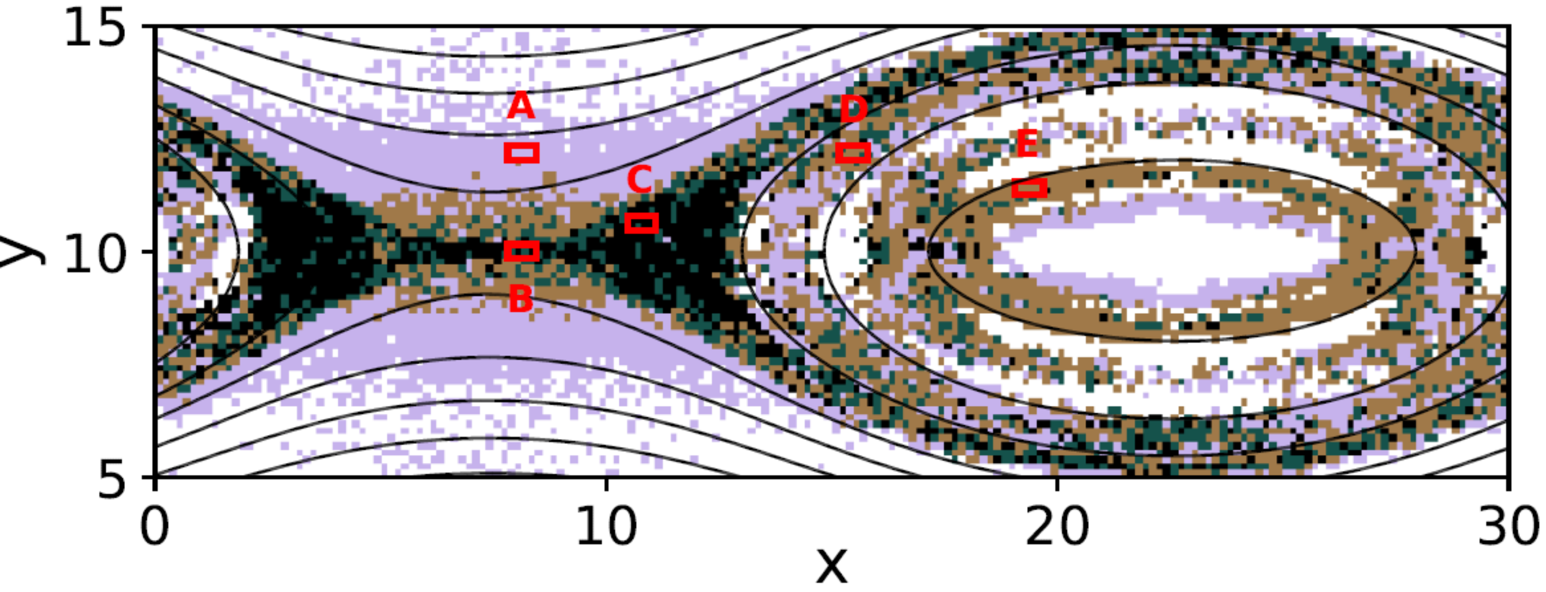
$$E_{drop} = \frac{E^{(K)}_{thermal}}{E_{thermal}}$$

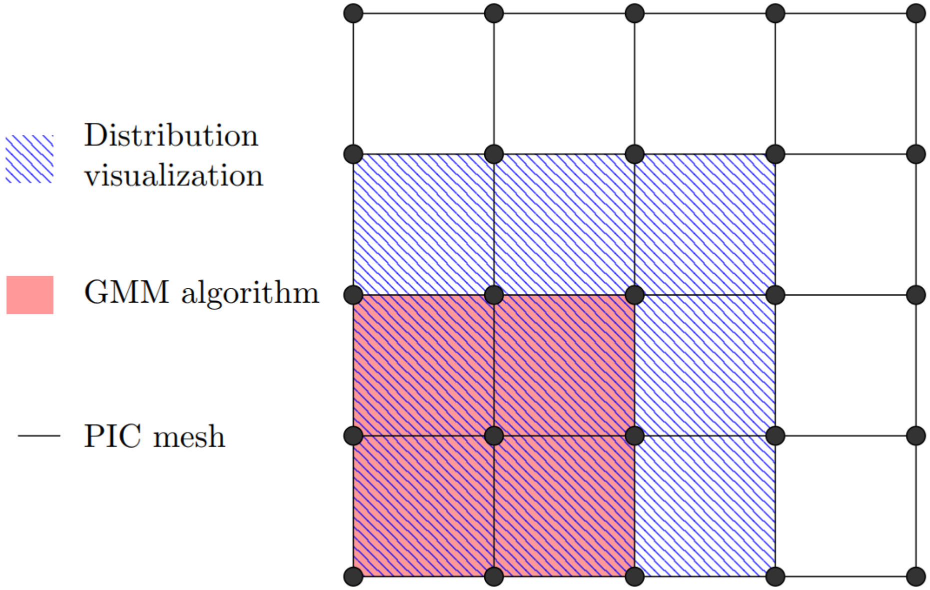
$$E^{(K)}_{thermal} = \frac{1}{2} \sum_{i=1}^{3} \sum_{k=1}^{K} w_k^2 \left[\boldsymbol{\sigma}_{\boldsymbol{k}}^2\right]_i$$

$$E_{dev} = \frac{E^{(K)}_{dev}}{E_{thermal}}$$

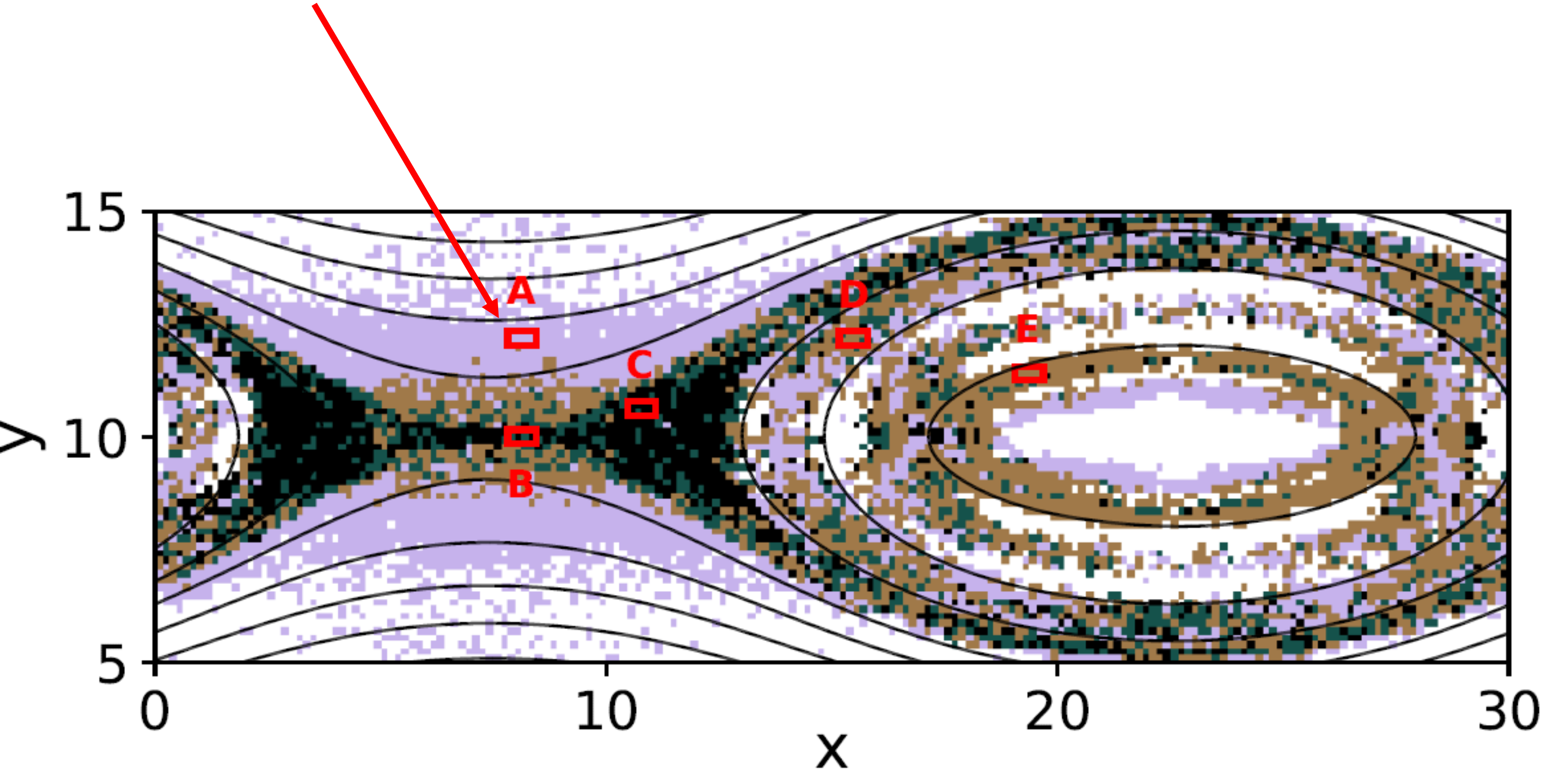
$$E^{(K)}_{dev} = \sum_{i=1}^{3} \left[\sum_{k=1}^{K} w_k \left(\boldsymbol{\mu}_{\boldsymbol{k}}\right)^2 - \left(\sum_{k=1}^{K} w_k (\boldsymbol{\mu}_{\boldsymbol{k}}) \right)^2 \right]_i$$

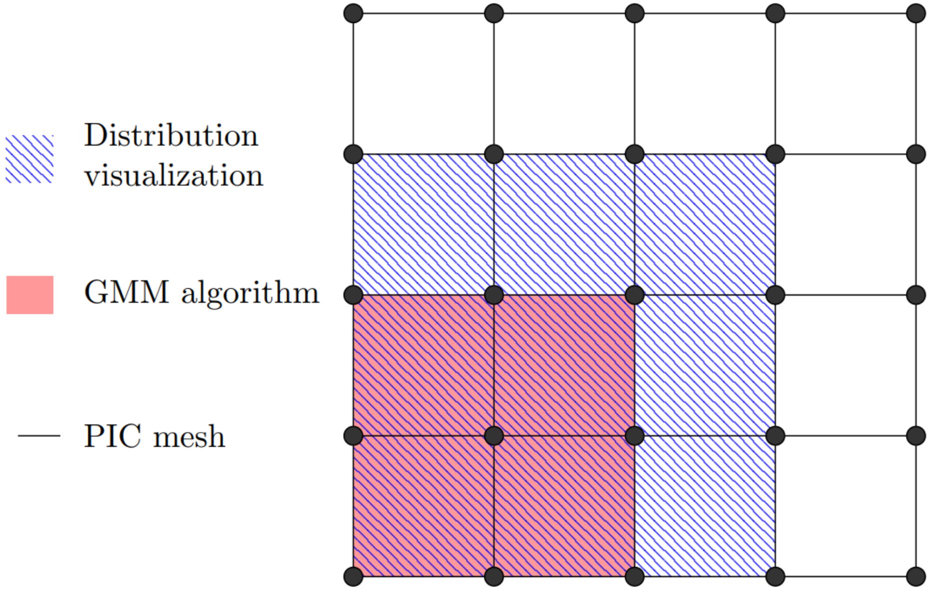
Closer look to the distributions



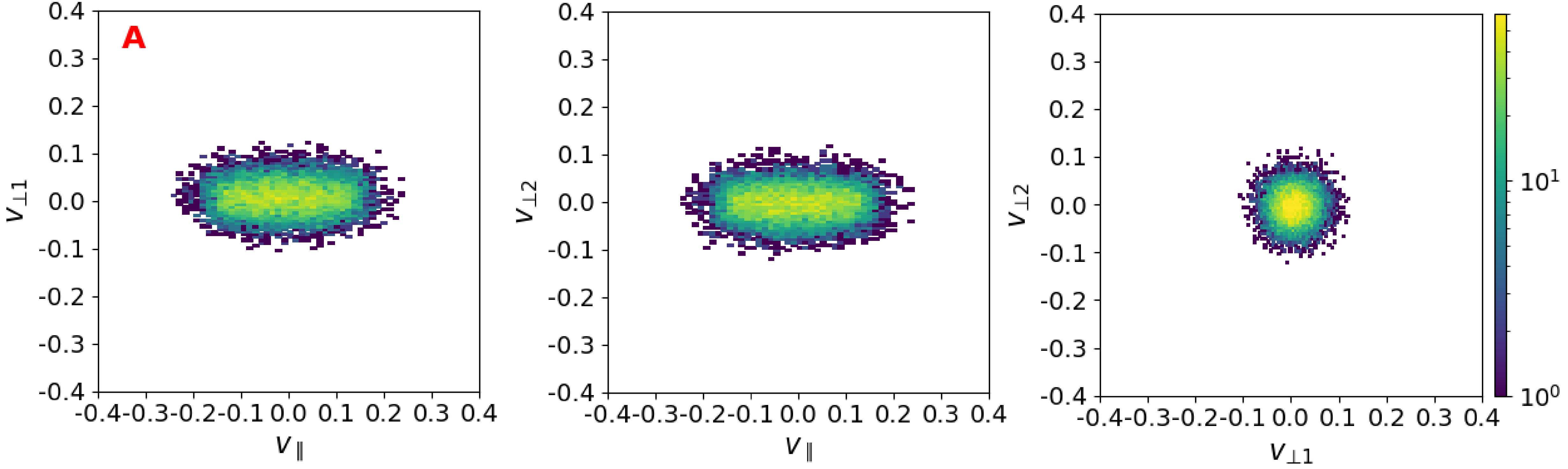


Closer look to the distributions

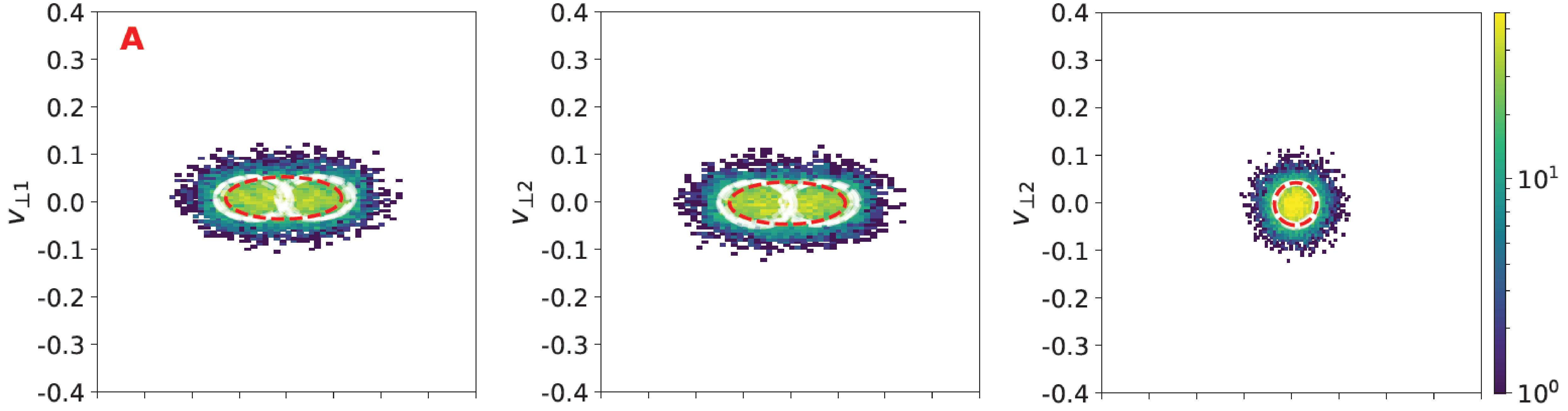




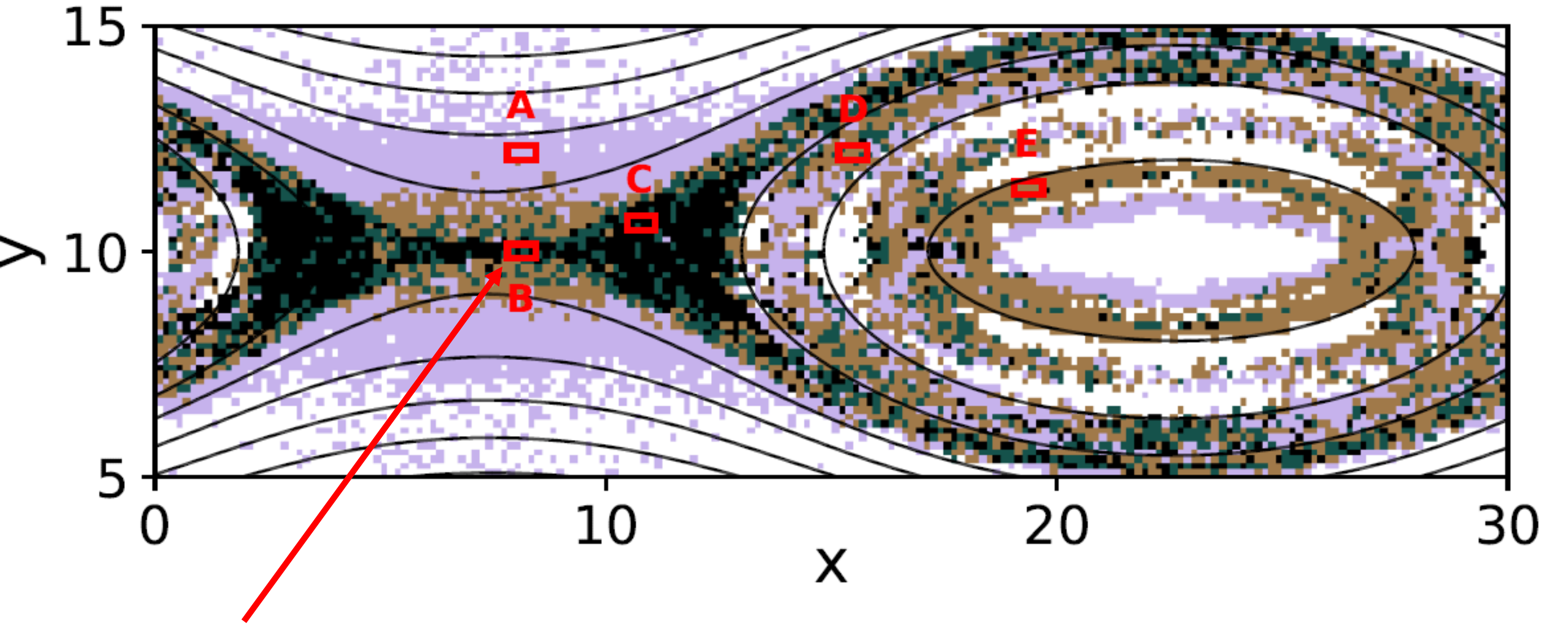
Closer look to the distributions

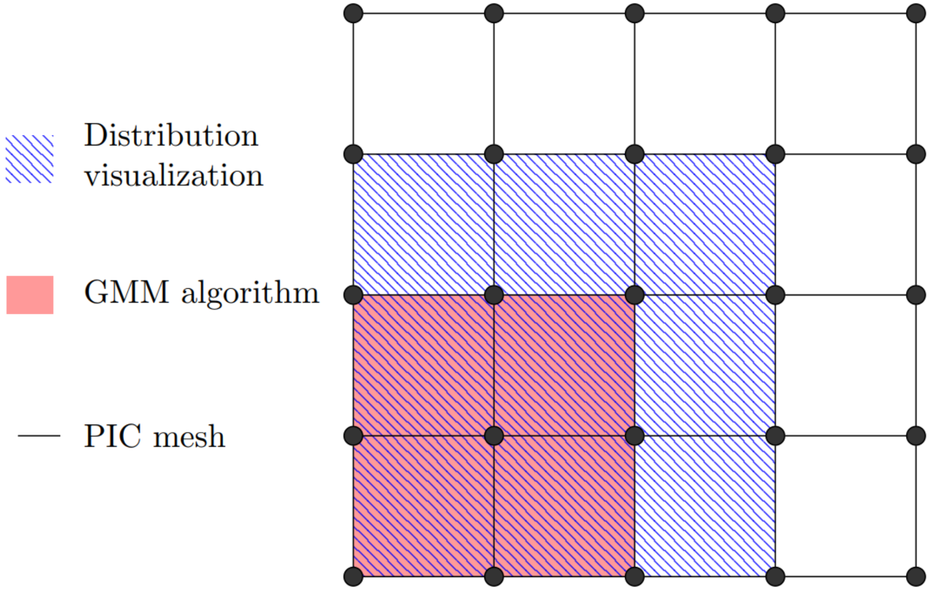


Closer look to the distributions

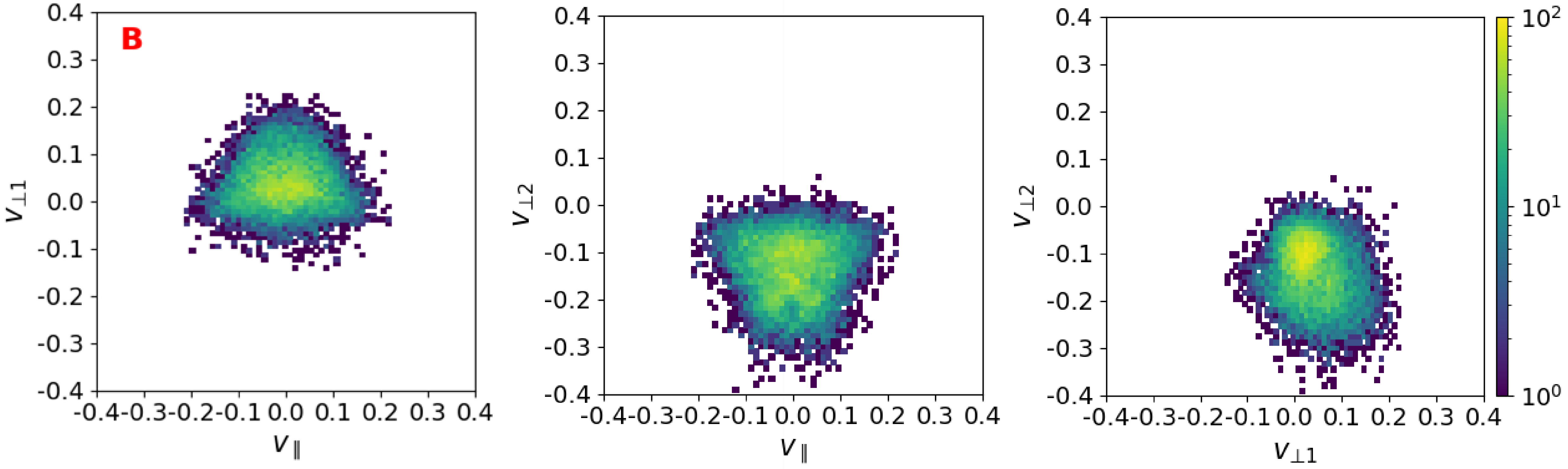


Closer look to the distributions

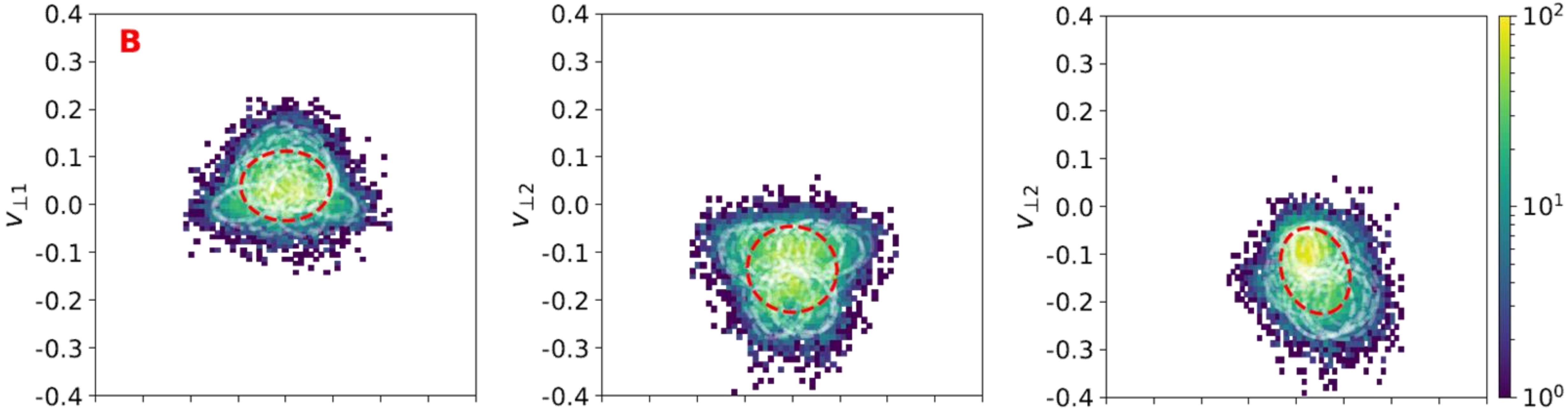




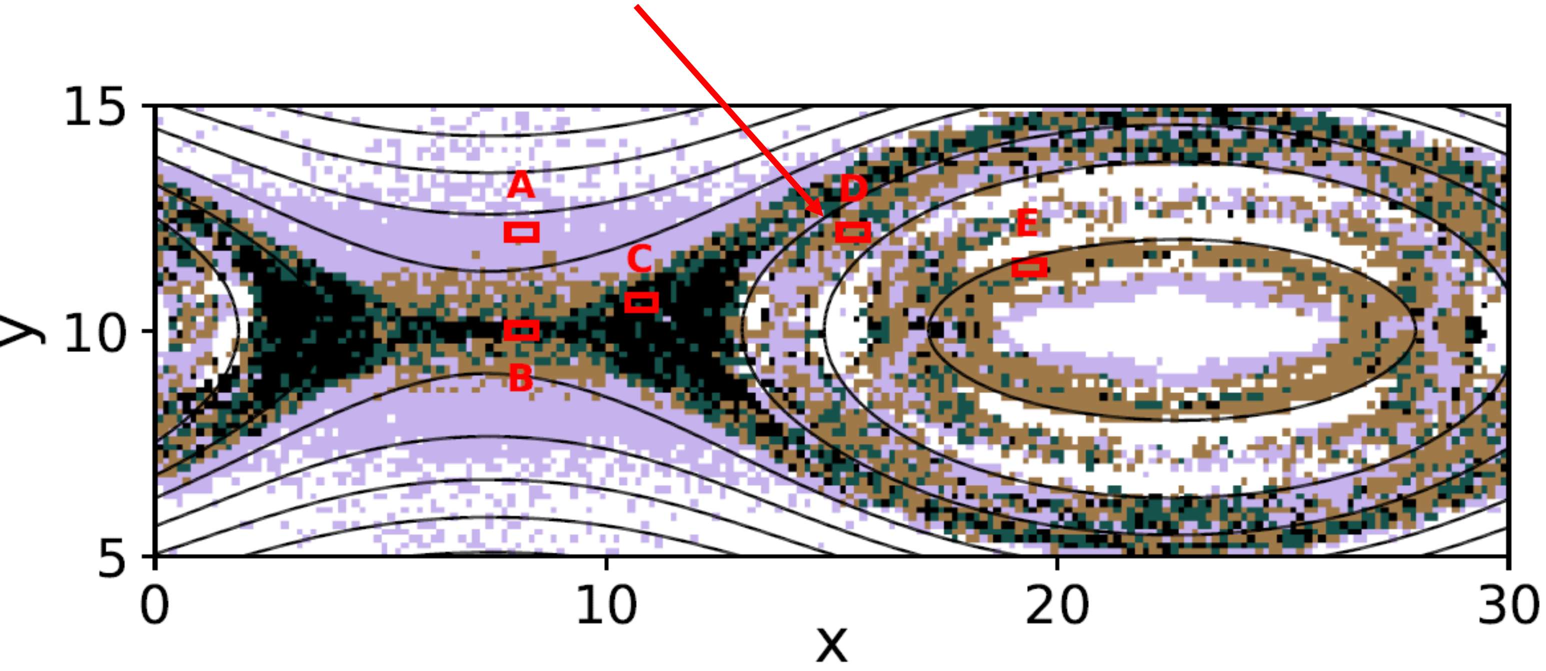
Closer look to the distributions

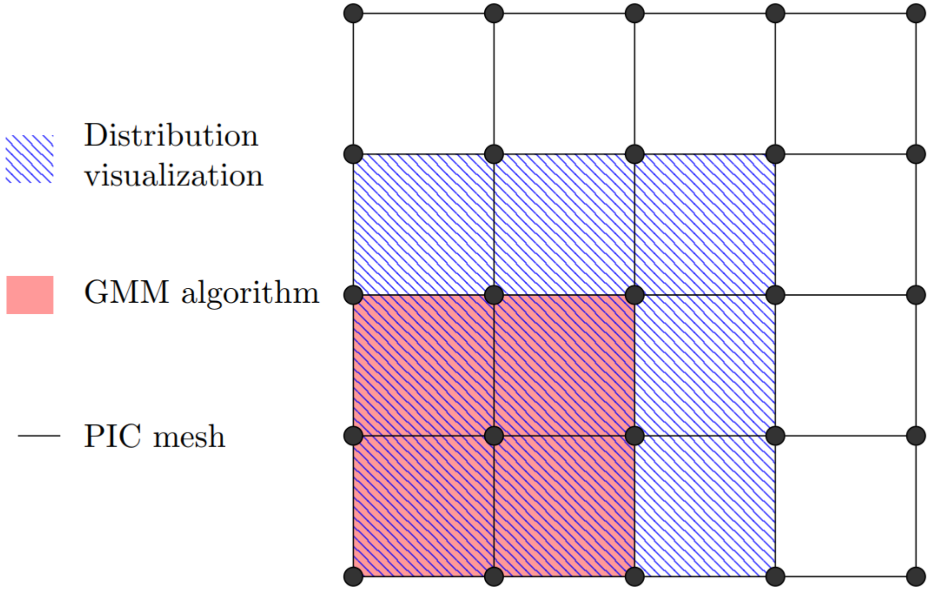


Closer look to the distributions

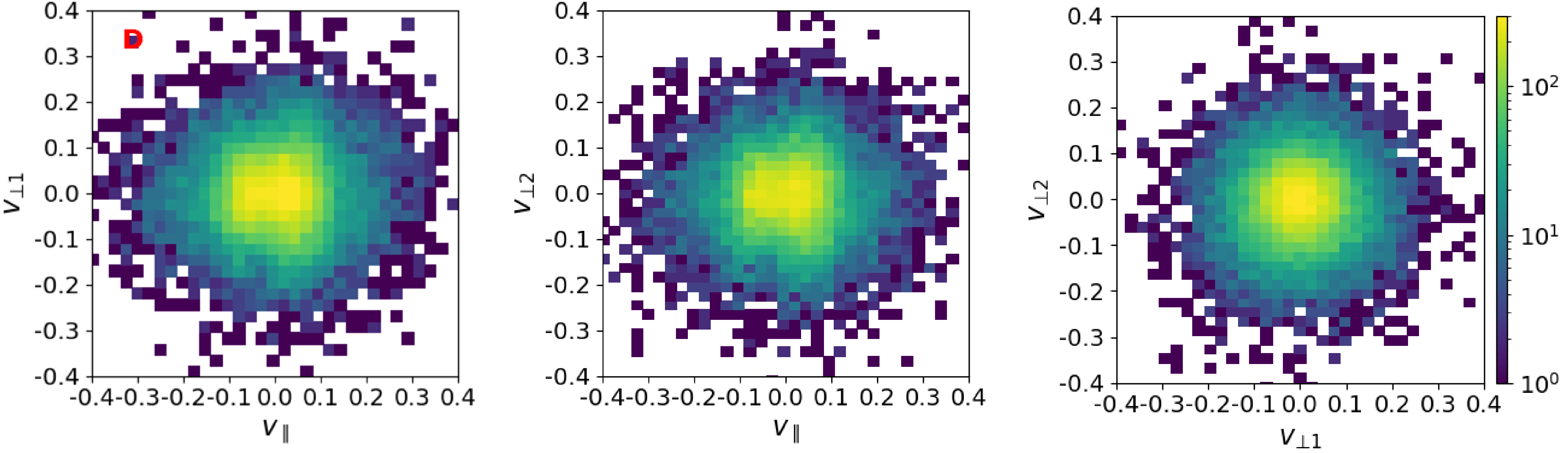


Closer look to the distributions

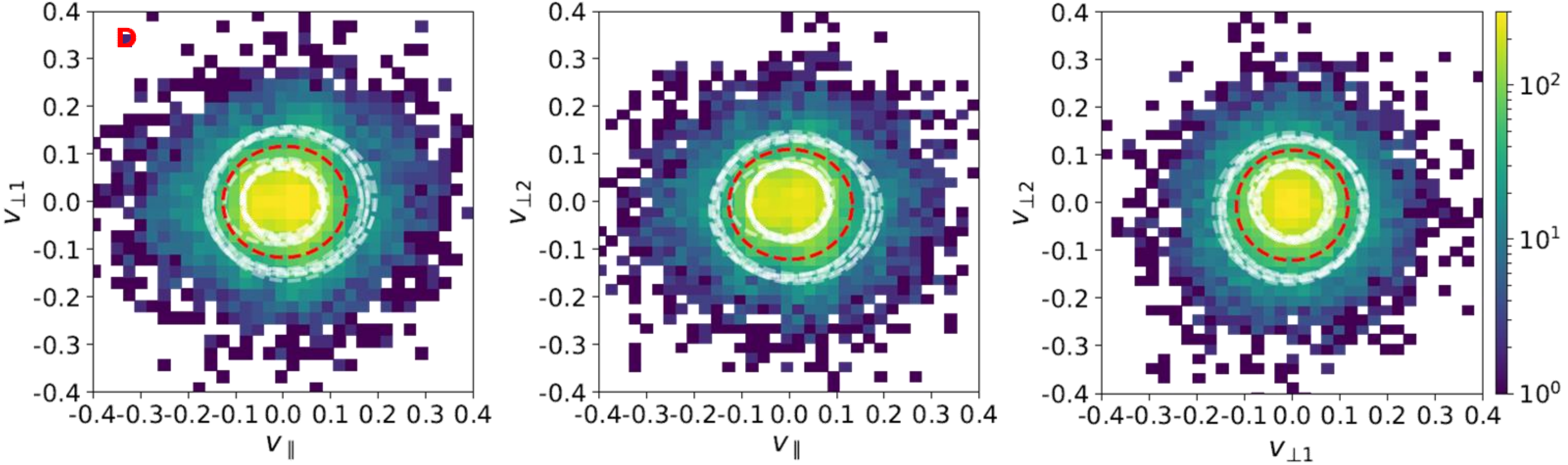




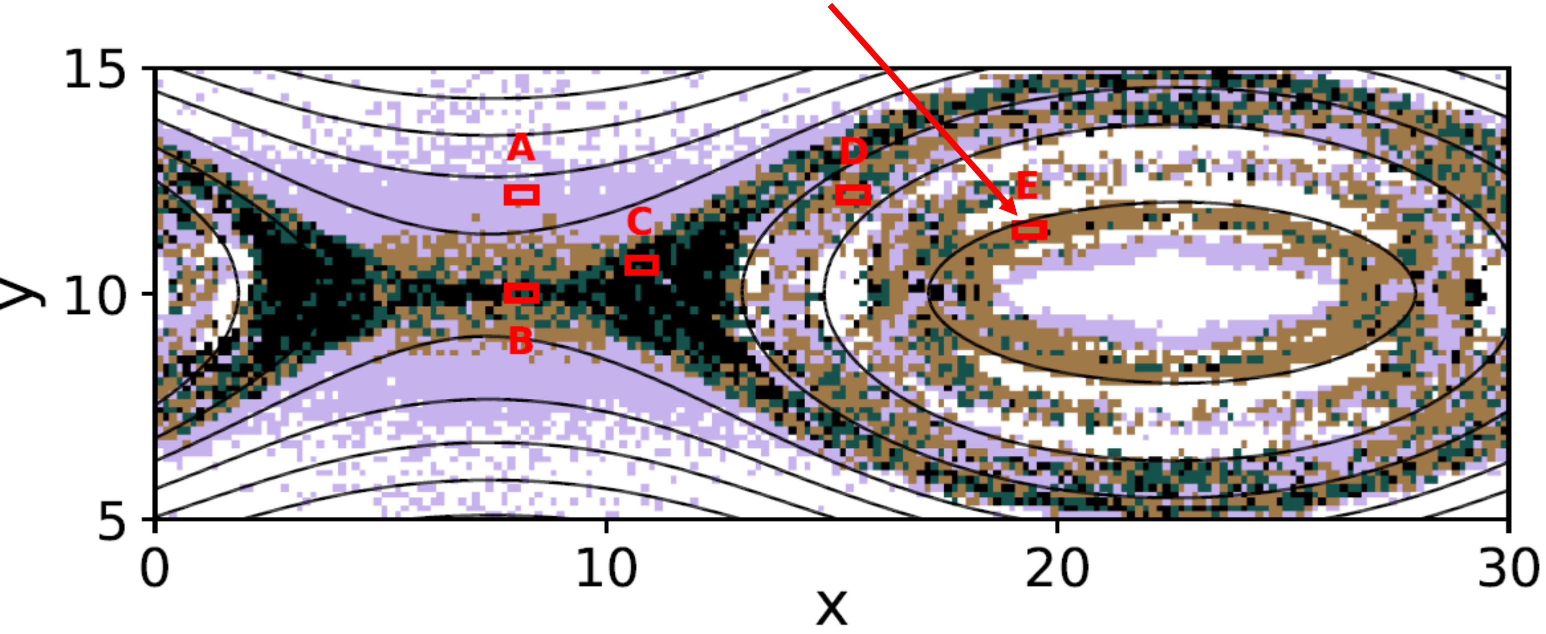
Closer look to the distributions

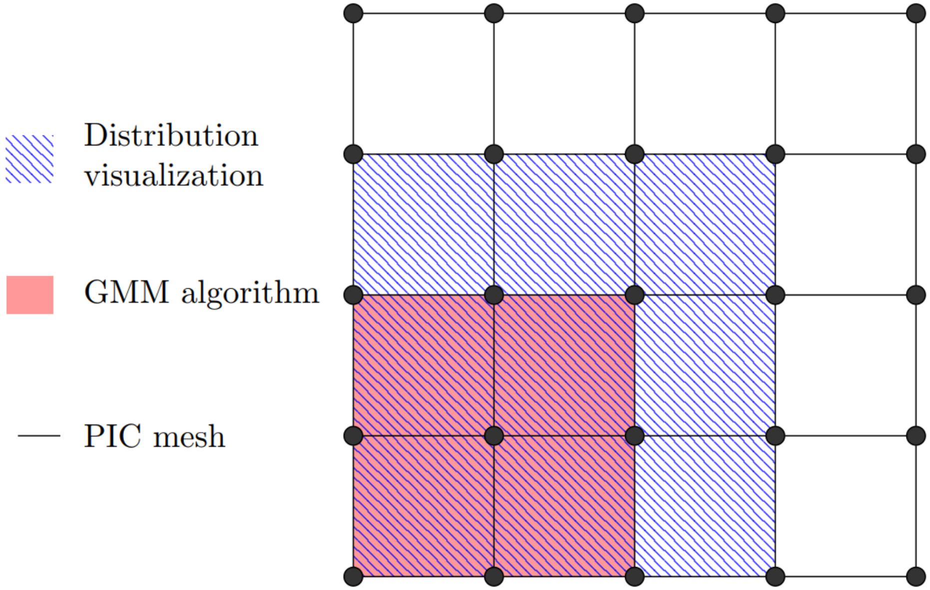


Closer look to the distributions

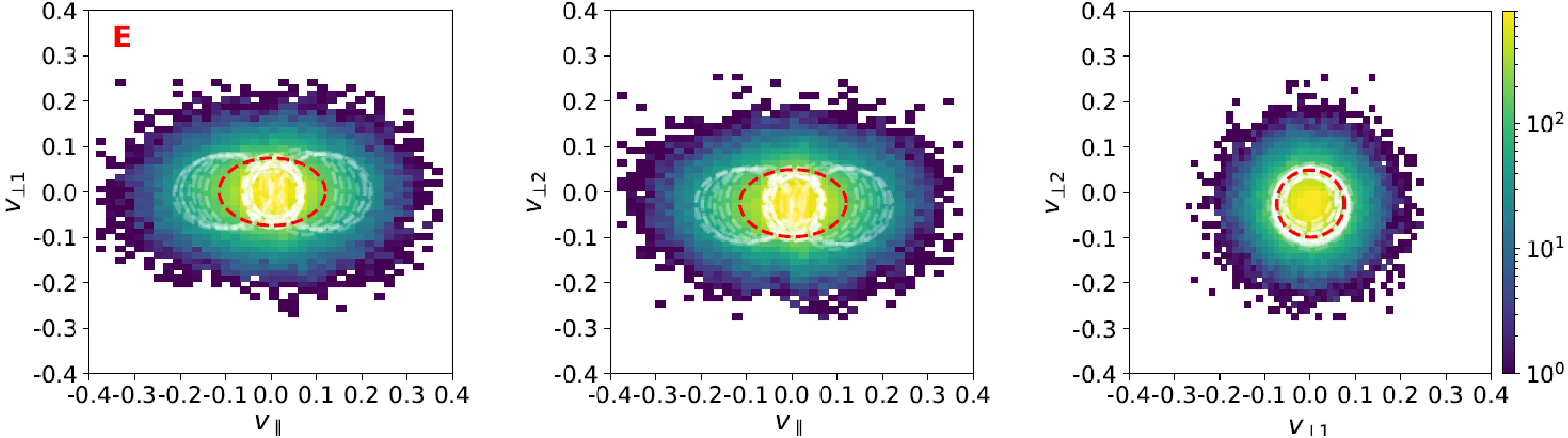


Closer look to the distributions





Closer look to the distributions



Next steps

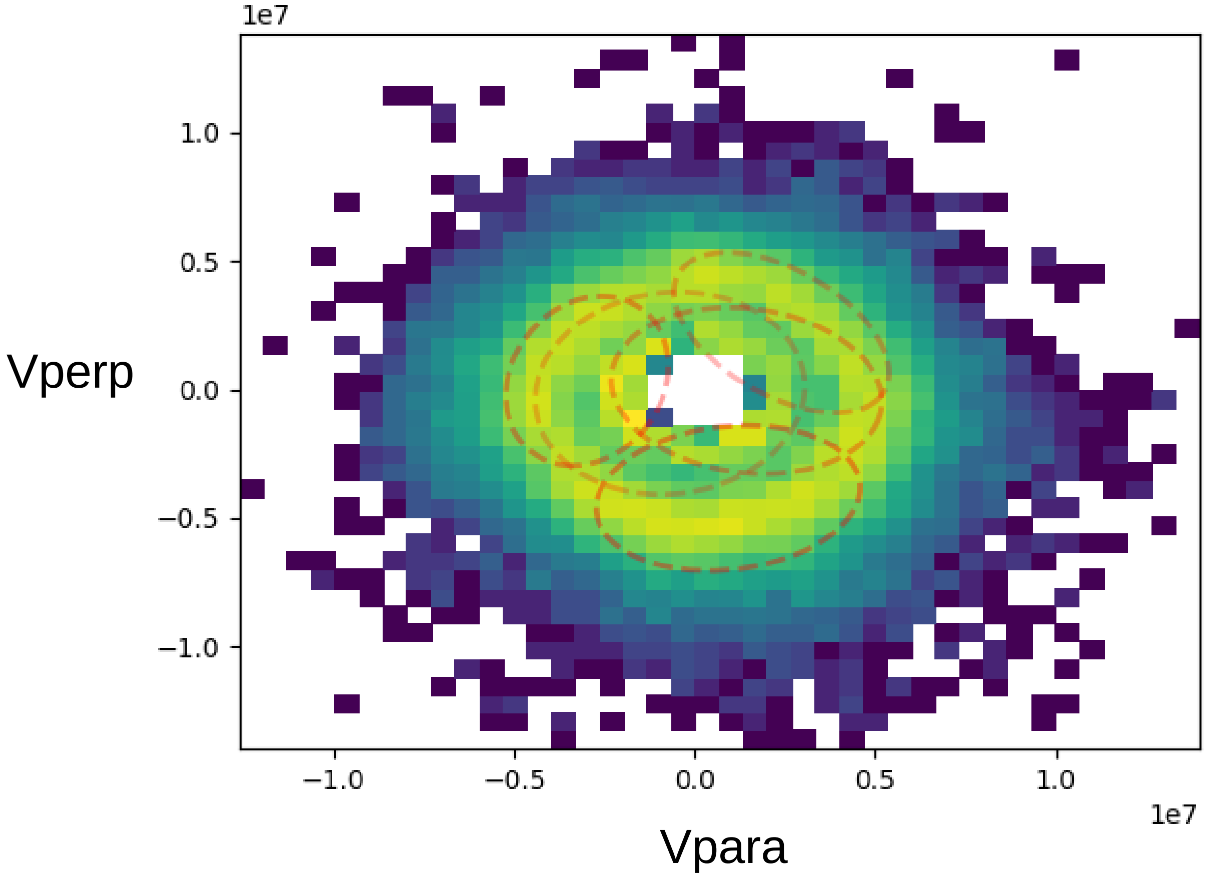
- **Simulation: focus on less documented cases**
  - 2D turbulent reconnection
  - 3D reconnection
  - Other kind of simulations

- **Observation: distributions from in situ space missions**
  - MMS data
  - Reconstruction of the particle sampling

- **Machine learning: potential improvement**
  - Convolutional methods and auto encoder
  - Dual problem with kernel method

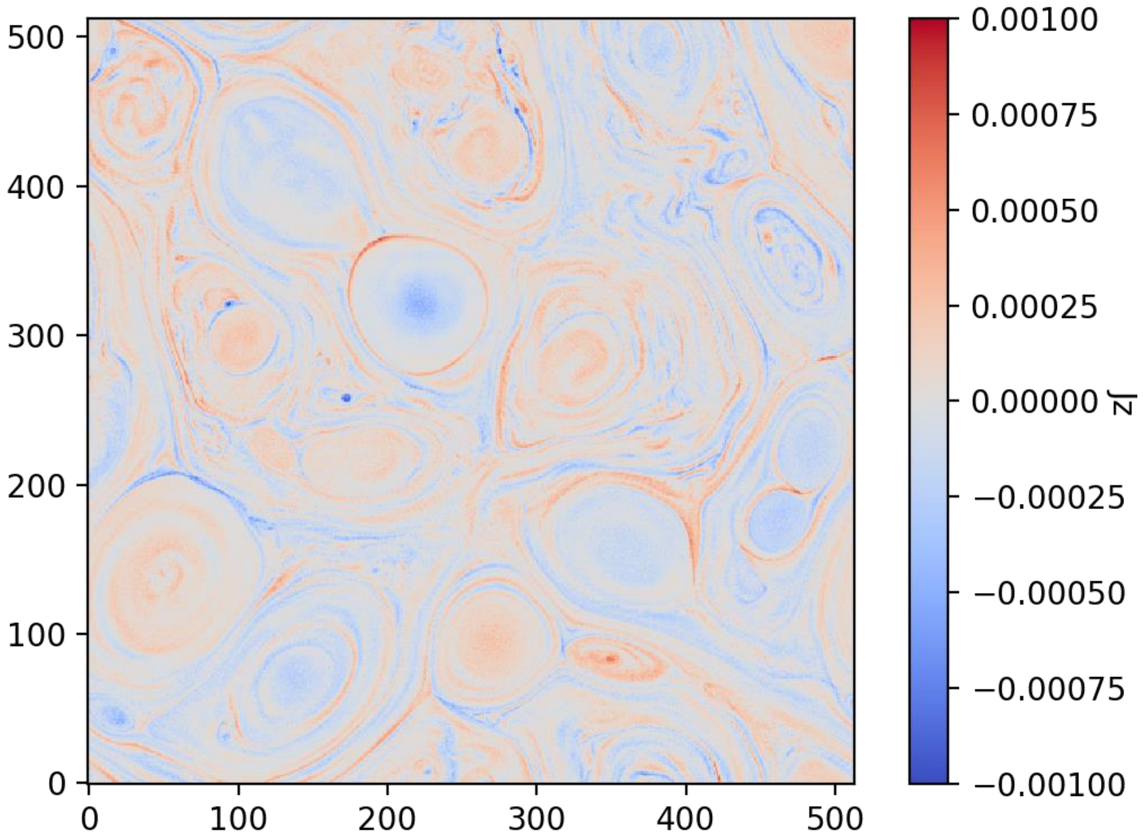
MMS observation: day side 2015

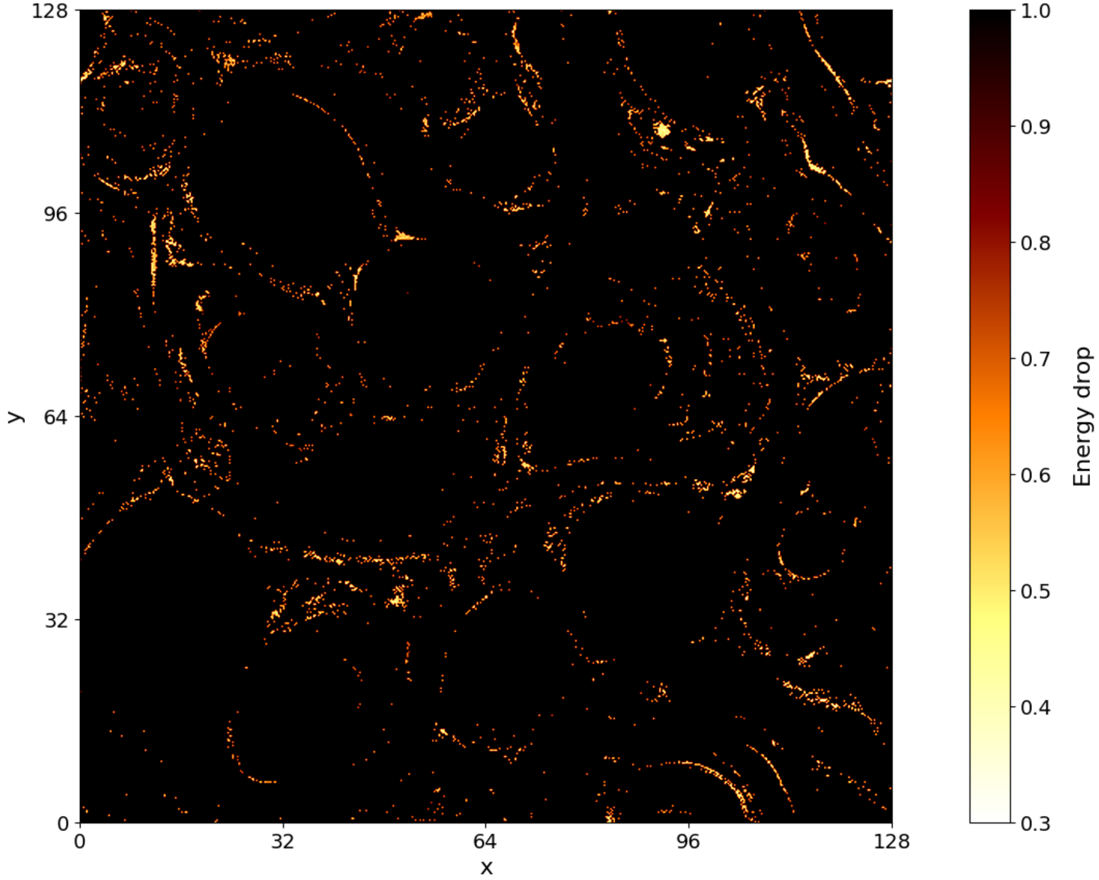
- **Reconnection event: 16 October 2015-13:07:02.235 (Burch et al.)**
  - Gap due to low energy channels
  - Complex preprocessing pipeline
  - Identify crescent shape
- **Auto encoder may help (**Olshevsky et al)
  - Extract 3D features
  - Use the PDF directly



Turbulent simulation

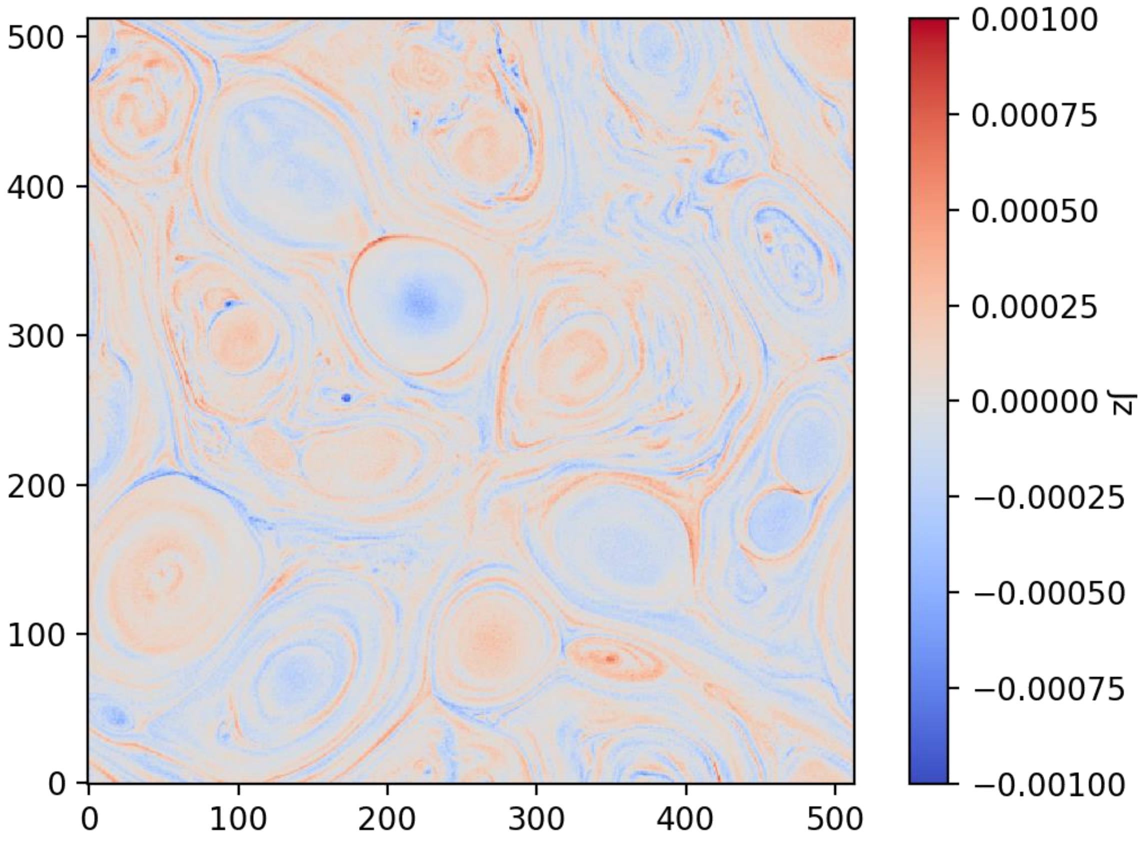
- **Identification of potential current layers**

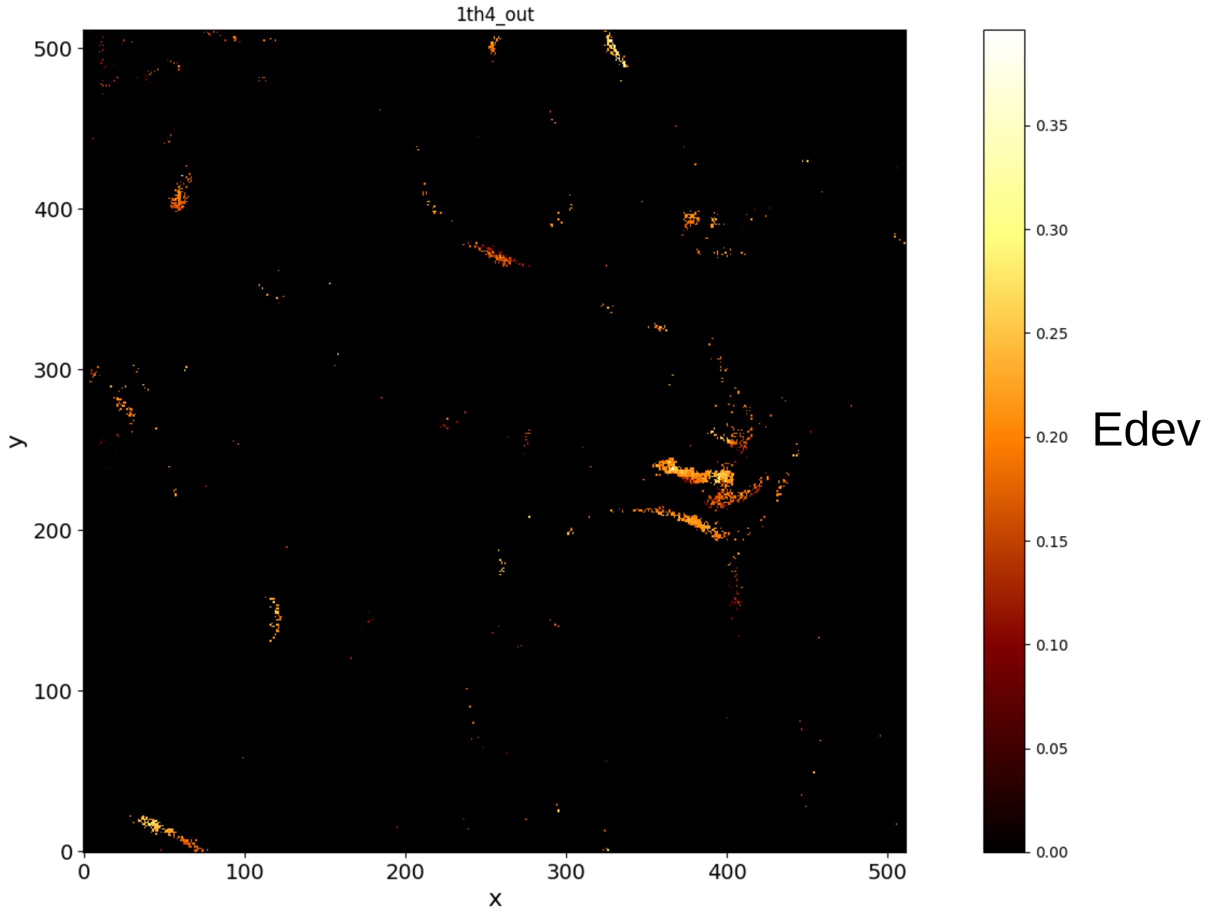




Turbulent simulation

- **Identification of potential current layers**





Conclusion

- **GMM is able to identify complex distributions in various cases**
  - Reconnection with weak and strong guide fields
  - Help to analyze and identify reconnection
  - No real physical interpretation and no unique solution

- **GMM as a start in applying ML on simulations and particles**
  - Other ML methods are considered
  - Auto encoder, SOM, etc.

References

Bishop, Christopher M. Pattern recognition and machine learning. springer, 2006.

Burch et al., “Electron-scale measurements of magnetic reconnection in space.”, 2016b, Science, vol. 352, no 6290, p. aaf2939

Dupuis, Romain, et al. "Characterizing magnetic reconnection regions using Gaussian mixture models on particle velocity distributions." The Astrophysical Journal 889.1 (2020): 22.

Garnier P. et al., Automatic detection of magnetopause reconnection diffusion regions

Gebru, Israel Dejene, et al. "EM algorithms for weighted-data clustering with application to audio-visual scene analysis." IEEE transactions on pattern analysis and machine intelligence 38.12 (2016): 2402-2415.

Markidis, S., & Lapenta, G. (2010). Multi-scale simulations of plasma with iPIC3D. Mathematics and Computers in Simulation, 80(7), 1509-1519.

Olshevsky et al., “Automated classification of plasma regions using 3D particle energy distribution”, submitted to JGR-Space physics

Shuster et al. “Highly structured electron anisotropy in collisionless reconnection exhausts”, 2014, Geophysical Research Letters, 41, 5389

Yeakel, K et al., Automatic Detection of Magnetospheric Regions around Saturn using Cassini Data, 2017

Acknowledgments

This work has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 776262 (AIDA, www.aida-space.eu).

Thank you for your attention

Contact: romain.dupuis@kuleuven.be

Gaussian Mixture Model

- **How to infer the best parameters ?**

$$l(\phi|\boldsymbol{X},\boldsymbol{Z}) = \sum_{i=1}^{n} \ln\left[\sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x_i}|\boldsymbol{\theta_k})\right]$$

- **Maximizing the likelihood estimation**
  - Non linear maximization problem
  - No closed form

$$\frac{\partial l}{\partial \boldsymbol{\mu_k}} = 0 \qquad \frac{\partial l}{\partial \boldsymbol{\Sigma_k}} = 0 \qquad \frac{\partial l}{\partial w_k} = 0 \;\; s.t. \sum_{k=1}^{K} w_k = 1$$

- **Need to find numerical local maximum: Expectation Maximization**

EM Algorithm

$$\gamma(z_k) = \frac{w_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\theta_k})}{\sum_{j=1}^{K} w_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\theta_j})}.$$

Expectation Step

$$\boldsymbol{\mu_k} = \frac{\sum\limits_{i=1}^{n} \gamma(z_{ik})\boldsymbol{x_i}}{\sum\limits_{i=1}^{n} \gamma(z_{ik})}, \ \forall k \in [1, \cdots, K],$$

$$\boldsymbol{\Sigma_k} = \frac{\sum\limits_{i=1}^{N} \gamma(z_{ik})(\boldsymbol{x_i} - \boldsymbol{\mu_k})(\boldsymbol{x_i} - \boldsymbol{\mu_k})^T}{\sum\limits_{i=1}^{N} \gamma(z_{ik})}, \ \forall k \in [1, \cdots, K]$$

$$w_k = \frac{1}{n}\sum_{i=1}^{n} \gamma(z_{ik}), \ \forall k \in [1, \cdots, K].$$

Maximization Step

Detection algorithm

Data: particles coordinates $\boldsymbol{X}$ and velocities $\boldsymbol{V}$, mesh Ω with n_x cells by n_y cells
Result: distribution function f_i

begin
  merge cells to increase the number of particles by subset, new mesh Ω' is of size $\frac{n_x}{r} \times \frac{n_y}{r}$
  for $i \in \Omega'$ **do**
    for $k = 1$ *to* $k = 6$ **do**
      $GMM_k = GMM(\mathbf{V_i}; k)$;
      $BIC_k = BIC(GMM_k)$;
    end
    $K = \arg\max_{k \in [1,6]} BIC_k$;
    $f_i = GMM_K$;
  end
end

Detection algorithm

Data: particles coordinates $\boldsymbol{X}$ and velocities $\boldsymbol{V}$, mesh Ω with n_x cells by n_y cells

Result: distribution function f_i

begin

  merge cells to increase the number of particles by subset, new mesh Ω' is of size $\frac{n_x}{r} \times \frac{n_y}{r}$

  for $i \in \Omega'$ **do**

    for $k = 1$ *to* $k = 6$ **do**

      $GMM_k = GMM(\mathbf{V_i}; k)$;

      $BIC_k = BIC(GMM_k)$;

    end

    $K = \arg\max_{k\in[1,6]} BIC_k$;

    $f_i = GMM_K$;

  end

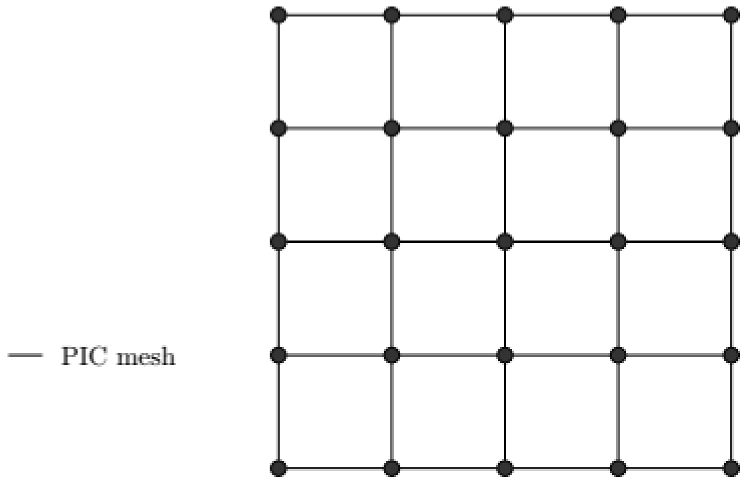
end

Detection algorithm

Data: particles coordinates $\boldsymbol{X}$ and velocities $\boldsymbol{V}$, mesh Ω with n_x cells by n_y cells

Result: distribution function f_i

begin
    merge cells to increase the number of particles by subset, new mesh Ω' is of size $\frac{n_x}{r} \times \frac{n_y}{r}$
    for $i \in \Omega'$ **do**
        for $k = 1$ *to* $k = 6$ **do**
            $GMM_k = GMM(\mathbf{V_i}; k)$;
            $BIC_k = BIC(GMM_k)$;
        end
        $K = \arg\max_{k \in [1,6]} BIC_k$;
        $f_i = GMM_K$;
    end
end



Detection algorithm

Data: particles coordinates $\boldsymbol{X}$ and velocities $\boldsymbol{V}$, mesh Ω with n_x cells by n_y cells

Result: distribution function f_i

begin

  merge cells to increase the number of particles by subset, new mesh Ω' is of size $\frac{n_x}{r} \times \frac{n_y}{r}$

  for $i \in \Omega'$ **do**

    for $k = 1$ *to* $k = 6$ **do**

      $GMM_k = GMM(\mathbf{V_i}; k)$;

      $BIC_k = BIC(GMM_k)$;

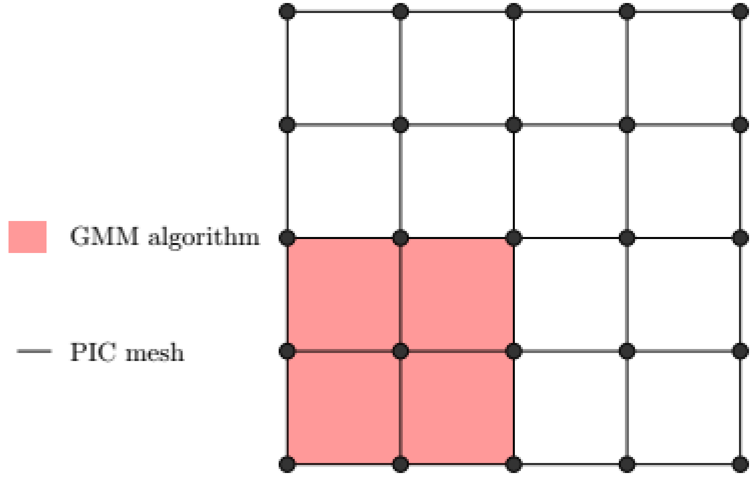
    end

    $K = \arg\max_{k \in [1,6]} BIC_k$;

    $f_i = GMM_K$;

  end

end



Detection algorithm

Data: particles coordinates $\boldsymbol{X}$ and velocities $\boldsymbol{V}$, mesh Ω with n_x cells by n_y cells

Result: distribution function f_i

begin

  merge cells to increase the number of particles by subset, new mesh Ω' is of size $\frac{n_x}{r} \times \frac{n_y}{r}$

  for $i \in \Omega'$ **do**

    for $k = 1$ *to* $k = 6$ **do**

      $GMM_k = GMM(\mathbf{V_i}; k)$;

      $BIC_k = BIC(GMM_k)$;

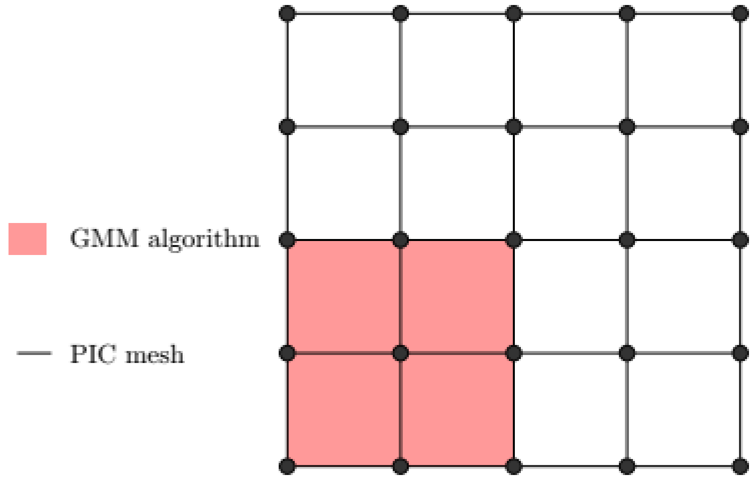
    end

    $K = \arg\max_{k \in [1,6]} BIC_k$;

    $f_i = GMM_K$;

  end

end



Detection algorithm

Data: particles coordinates $\boldsymbol{X}$ and velocities $\boldsymbol{V}$, mesh Ω with n_x cells by n_y cells

Result: distribution function f_i

begin

  merge cells to increase the number of particles by subset, new mesh Ω' is of size $\frac{n_x}{r} \times \frac{n_y}{r}$

  for $i \in \Omega'$ **do**

    for $k = 1$ *to* $k = 6$ **do**

      $GMM_k = GMM(\mathbf{V_i}; k)$;

      $BIC_k = BIC(GMM_k)$;

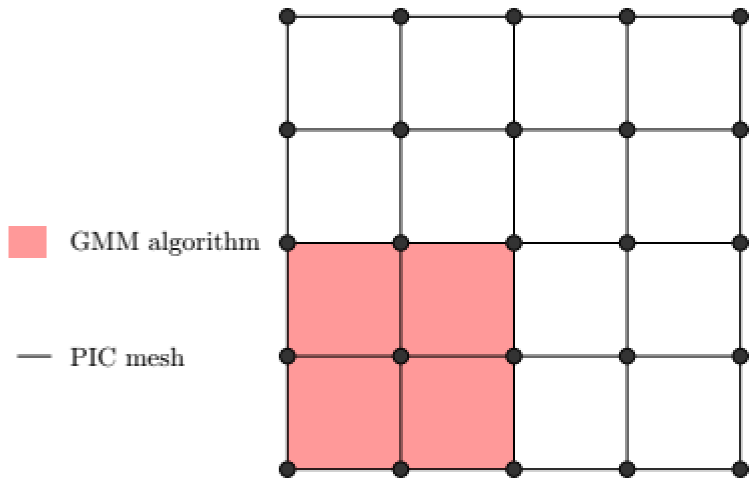
    end

    $K = \arg\max_{k \in [1,6]} BIC_k$;

    $f_i = GMM_K$;

  end

end



Detection algorithm

Data: particles coordinates $\boldsymbol{X}$ and velocities $\boldsymbol{V}$, mesh Ω with n_x cells by n_y cells
Result: distribution function f_i

begin
  merge cells to increase the number of particles by subset, new mesh Ω' is of size $\frac{n_x}{r} \times \frac{n_y}{r}$
  for $i \in \Omega'$ **do**
    for $k = 1$ *to* $k = 6$ **do**
      $GMM_k = GMM(\mathbf{V_i}; k)$;
      $BIC_k = BIC(GMM_k)$;
    end
    $K = \arg\max_{k \in [1,6]} BIC_k$;
    $f_i = GMM_K$;
  end
end