Technical Report, A Kernel-Embedding Framework for Inferring Generic Electron Energy Distributions from Spectroscopic Diagnostics (FA9300-20-F-9801)

Hal J. Cambier (CTR), David L. Bilyeu

Air Force Research Laboratory's In-Space Propulsion Directorate
1 Ara Drive
Edwards AFB, CA 93524

March 18, 2022

Contents

List of Figures

1 SUMMARY

Inferring more exotic electron energy distribution functions (EEDFs) from emission or absorption spectra faces potential physical limitations, and complexity challenges, but is crucial for meeting certain AFRL objectives. A forward model plus Markov Chain Monte Carlo (MCMC) over parameters *influencing* the EEDF forms a common strategy for confronting the problem. This strategy automatically incorporates physical constraints, but involves situation-specific knowledge (or *assumptions*).

While we anticipate utilizing problem-specific knowledge, and any other available diagnostics in ultimate use cases, approaching the problem from the agnostic limit proved fruitful. Besides letting us examine model and diagnostic assumptions, it spurred developing a more flexible EEDF representation, and framework for inference.

Specifically, we generalized EEDF shape through splines or weighted basis functions in cumulative-distribution-function (CDF) space, and performed inference via data-driven kernel-embedding techniques – as well as a more conventional method combining annealing with MCMC. The kernel-embedding approach accomodates problem-specific and agnostic EEDFs, facilitates faster inference, and can complement more conventional MCMC. Crucially, it also provides a handle on the *messiness* of the problem.

2 INTRODUCTION AND BACKGROUND

The swiftness of momentum and energy exchange between electrons often keeps the EEDF close to a Maxwell-Boltzmann (MB) distribution. However, Air Force applications such as Hall effect thrusters, directed energy weapons, and plasma-assisted combustion defy this tendency by introducing complications like strong driving (directly via fields, indirectly through excited populations), low electron densities, and rapid variability. Other parametric distributions (e.g. generalized Druyvesteyn, Margenau) sometimes suffice to capture the resulting EEDFs, but not always (e.g. strong impact transition features, anisotropy beyond a bi-Maxwellian approximation).

Inferring the EEDF from emission and/or absorption spectroscopy presents significant – sometimes insurmountable – challenges as it entails viewing the EEDF through the lens of collisional-radiative physics, and radiation transport (RT) for more optically-thick systems. The CR physics may be complex to unravel, plagued with uncertainties, or uninformative[1].

More direct diagnostics exist, but also face various challenges. Some perturb the EEDF they are meant to measure, or suffer low signal-to-noise. At a more practical level, they may be expensive, bulky, and require proximity to the plasma (which may also erode physical probes). For studying the EEDF in a small device, remotely, or developing lightweight in-flight diagnostics for satellites, spectroscopy may be the only option. Of course, spectroscopy can also complement other diagnostics when they are available, further motivating techniques to extract more information from this channel.

As noted, common paths to handle the complexity of spectroscopic EEDF inference include simplifying assumptions about the EEDF, or MCMC with higher fidelity models that shift inference to problem-specific inputs. At the very least, testing whether EEDFs 'near' the model also explain the observations indicates how incisive the diagnostic, or prior is. Besides supporting broader priors, efficiently exploring wilder perturbations can help plan experiments. For example, we would like to feed a simple model relevant species densities, lines observed, trapping factors, etc. and chart where distinguishing various EEDF features is feasible. A more controlled, "perturbative" perspective of course allows more targeted resolution. We began by considering arbitrary (quasi-static) EEDFs, but connect it to perturbative perspective in the kernel embedding framework.

[1] For a simple example: a species with accurate cross sections would still not provide a useful "tracer" if the EEDF is so relatively 'cold' that no meaningul excited state population exists

3 METHODS, ASSUMPTIONS, AND PROCEDURES

We first review how we represent more generic EEDFs in §3.1. We briefly review the transitional MCMC (TMCMC) method in §3.2 as it offers complementary or fallback capability, and enters later discussion. We then focus on the approach using basic kernel-embedding of conditional distributions in §3.3.

3.1 Representation/Parameterization

While we avoid specific, parametric models in usual sense, some finite representation – i.e. parameterization – remains necessary. We explored several formulations, which ultimately share a compositional nature so that some techniques may be recycled for very different problems. Before launching into these variations and their nuances it bears emphasizing that our preferred kernel-embedding (KE) approach does not hinge on these parameterizations, but they still prove a useful problem-agnostic proxy for feasibility studies mentioned above. They may also help interpolate, and perturb 'referece' EEDFs to flesh out space supporting the prior (*when* new observables can be estimated).

We begin with the basic histogram and its connection to a cumulative distribution function (CDF) representation. The CDF cleanly separates the EEDF scale from its shape, which helps immensely when strong prior or observational constraints on total electron density exist, or when organizing a feasibility study. In this framework, the range of allowed electron energy distribution funciton (EEDF) shapes are all monotonic paths from the origin to some cutoff energy, and total number density. Fig. 1 illustrates the connection for a basic histogram, and introduces the interval coordinates α.

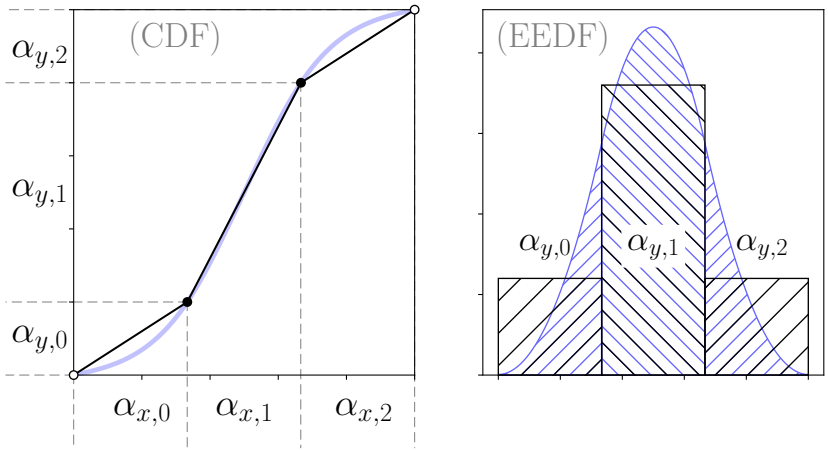

Figure 1: Left panel shows piecewise-linear (thin, black) and exponential-tension spline (thick, faint blue) interpolation in CDF space, and corresponding representations in EEDF space, as well as their connection with interval 'coordinates' α on each axis. The effective degrees of freedom are two if the interval grid (and inner control points) are fixed on one axis, and four if both axes are free.

Note that fixing the α_x for a piecewise-linear CDF just yields a conventional histogram representation with α_y the relative bin counts, which must obey a sum constraint. Varying α_x instead captures small percentiles in the high-energy tail more consistently. Varying both is possible, but formally introduces some redundancy.[2]

[2] As a simple example suppose a portion of the EEDF is roughly constant corresponding to a constant-slope line segment in the CDF. There are many ways to place control points along the line that produce the same EEDF. At coarser discretizations, the issue is less severe, while flexibility becomes more crucial.

To recycle standard MCMC machinery typically written with unbounded real coordinates in mind, and sidestep the challenge of tracking the mass constraint, we may employ additive logratio transformations:

$$u_r \equiv \text{alogr}(\alpha) = \begin{pmatrix} \frac{\alpha_1}{\alpha_0} & \frac{\alpha_2}{\alpha_0} & \dots \end{pmatrix}^\intercal \tag{1a}$$

$$\alpha \equiv \text{alogr}^{-1}(u_r) = \begin{pmatrix} \frac{1}{1+\sum \exp[u_k]} & \frac{\exp[u_1]}{1+\sum \exp[u_k]} & \dots \end{pmatrix}^\intercal \tag{1b}$$

where here the first component, α_0, of the component vector was chosen as the reference value, but one could choose a different reference component. Note the dimension of u is one less than α reflecting the true degrees of freedom. Obviously the transformation is undefined if the reference component is zero (or numerically small). The PhD thesis of Bear[1] nicely summarizes some strategies for dealing with true zeros, but for this work a numerical floor sufficed. Variable transformations for bounded-support scalars like total density, heavy species' densities, and more general linear constraints also exist.

Compositional coordinates naturally apply to weighted sums of shape functions, and we did so with emphasis on I-spline functions (M-splines in EEDF space). These, and appropriate choices for functions can automatically enforce feasibility (CDF monotonicity / EEDF positivity, boundary conditions, etc.). However, this approach can still require more degrees of freedom to reproduce smoother EEDFs, which is clearly undesirable.

Exponential tension splines[2, 3], particularly in logarithmic space, proved a convenient solution to our various objectives. For the linear case, enforcing zero density of states at zero energy implies zero CDF slope at the origin, and the typical resulting inflection produces large jumps in the EEDF spline (or demands many points of low value for the CR response). Logarithmic space is better suited to capture the CR response, and steep fall-off in the EEDF. For the low-energy cutoff: decay, or constant-slope (self-similar) boundary conditions are natural.

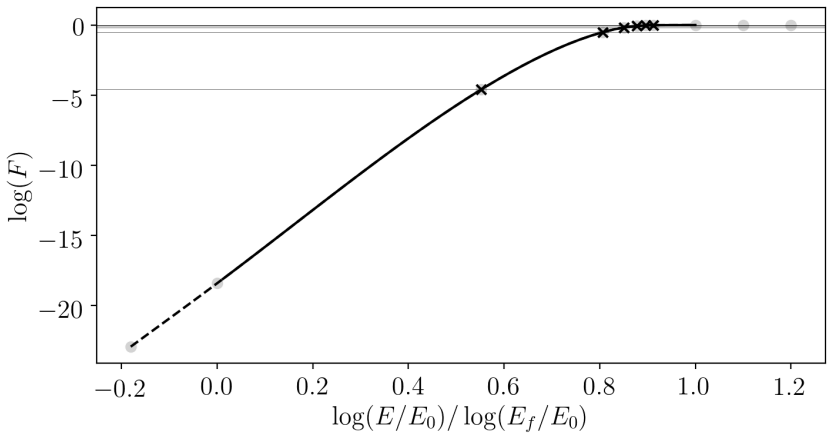

Figure 2: Exponential tension spline in log-log space with a fixed percentile grid (horizontal lines) that focuses on the high energy tail. The "Xs" mark control points that 'slide' along the horizontal lines within the allowed energy range. Due to the nature of the tension adjustment algorithm, ghost points past the cutoffs prove helpful at enforcing boundary conditions without introducing oscillations. The dashed segment highlights the self-similar boundary condition.

3.2 Transitional Markov Chain Monte Carlo (TMCMC)

As stated above, we applied a type of adaptive-annealing Monte Carlo method known as Transitional Markov chain Monte Carlo[4, 5, 6]. Parallel tempering MCMC also tackles multimodality, and lends itself to trans-dimensional parameter inference, but TMCMC offers advantages for our immediate interests.

Since our parameterization organizes the search over EEDF shapes, we can cast an appropriately wide particle "net" at the first annealing stage. Anticipating MCMC methods as a backup for inference feasibility studies,

TMCMC was also attractive for providing information even for partially-complete runs, i.e. initial sweeps over a grid of conditions with computational cost ceilings. The effective temperatures reached in each case would already indicate relative difficulty of inference. The picture of the posterior, and extent of multimodality, would also appear in a more gradual, but consistent way.

For convenience and clarity, we include the pseudocode (alg.1) and a brief description of the method below.

Algorithm 1: Transitional Markov Chain Monte Carlo

Given static observations y;
Initialize $\beta_0 = 0$, particle $\theta_p \sim$ prior $p(\theta)$, , $S_0 = 1$...;
for $j = 1, \ldots,$ `max_tempering_steps` $- 1$ **do**
  For grid of candidate ${\beta_j}^{(c)} \in (\beta_{j-1}, 1]$ get partial plausibility updates

$$w_{p,j}^{(c)} = (\mathcal{L}(y|\theta_p))^{({\beta_j}^{(c)} - \beta_{j-1})}$$

  and relative weights $\tilde{w}_{p,j}^{(c)} = w_{p,j}^{(c)} / \sum_p w_{p,j}^{(c)}$;
  Compute coefficient of variation over candidates' relative weights: COV($\tilde{w}^{(c)}$);
  Pick $c, \beta_j \leftarrow \beta_j^{(c)}$ for which COV($\tilde{w}^{(c)}$) is closest to some threshold (usually 1) without going over;
  Record $S_j = \frac{1}{N_p} \sum w_p$;
  (Might prepare adaptive proposal distributions *here*, before resampling);
  Resample-with-replacement θ_p according to relative weights $\tilde{w}_p$;
  For each particle, run standard Metropolis-Hastings MCMC (typically with number of steps equal or greater than number of resampling clones);
  Update evidence estimate $\hat{p}(y) \leftarrow S_j \cdot \hat{p}(y)$;
  if $\beta_j = 1$ **then**
    `return()`;
  end
end

Starting from the prior, this targets an intermediate power of the likelihood distribution. Greedier candidate $\beta^{(c)}$ have lower mean weight and coefficient of variation (COV). Very low COV means many of the unique parameters will be dropped in the resampling, corresponding to excessive exploitation versus exploration; particles would struggle to re-explore the space sufficiently fast under the new $\mathcal{L}^{\beta}$, and even dominant modes might be missed by quenching β so suddenly.

In case the sustained proposal adaptivity is alarming given the usual mantra that adaptivity must diminish in MCMC, remember that rule is to protect stationarity for a single particle chain. Here, even if the proposal distribution eventually traps particles in various posterior modes, *given* sufficient particles to catch all the modes, then the overall ensemble still emulates the complete Markov chain. In our problem, a few modes may still be informative, but extreme multimodality translates to "non-feasible", so we only need sufficient particle count to catch such extreme scenarios qualitatively.

3.3 Kernel-embedding Conditional Distributions

The work reported here most closely followed conditional distribution kernel embedding as practiced in Song et al.[7], and Fukumizu et al.[8]. We very briefly introduce the basic concept in §3.3.1 for readers less familiar with kernel embeddings, before covering more practical aspects and implementation in §3.3.2.

3.3.1 Kernel-Embedding Background

We start from a more familiar, explicit feature space perspective in a setting with continuous real variables[7, 8]. Consider some nonlinear feature map $\phi(x)$ and embedding of a function $f(x)$ via:

$$\int f(x') \begin{pmatrix} \phi_0(x') & \phi_1(x') & \dots \end{pmatrix} \mathrm{d}x'$$

where the ϕ_i could be terms of a polynomial expansion, indicator functions (i.e. bins), more detailed shape functions, etc. This relates the function to some point in a high dimension space, and progressively richer feature spaces effectively capture more moments by mapping functions to more unique points. More formally though, we choose feature spaces that are Hilbert spaces which satifisfy the reproducing property:

$$\int f(x') \left(\phi(x') \cdot \phi(x)\right) \mathrm{d}x' = f(x)$$

If x represents a randomly-distributed variable, we can think of embedding the probability distribution governing x as embedding the measure, or some weighting function

$$\mu_X = \mathbb{E}_X[\phi] = \int \phi(x') \mathrm{d}P_X(x') = \int \phi(x') w(x') \mathrm{d}x'$$

Expectations of $f(X)$ then correspond to dot products in feature space

$$\mathbb{E}_X[f] = \int f(x) (\phi(x) \cdot \phi(x')) \, \mathrm{d}P_X(x') = \langle f, \mu_X \rangle \tag{2}$$

which highlights the connection to the common "kernel trick" in machine learning that replaces feature vectors, and their dot products, with kernel functions

$$\langle \phi(x_i), \phi(x_j) \rangle \leftrightarrow K(x_i, x_j) \ , \qquad \phi(\cdot) \leftrightarrow K(\cdot, x)$$

The connection between an embedding satisfying the reproducing property, and a spectral decomposition becomes more transparent written in terms of the kernel:

$$f(x) = \int f(x') \begin{pmatrix} \phi_0(x') & \phi_1(x') & \dots \end{pmatrix} \mathrm{d}x' \leftrightarrow \int f(x') K(x', x) \mathrm{d}x'$$

Joint distributions are embedded in a similar fashion:

$$\mathcal{C}_{XY} = \int \phi_X(x') \otimes \phi_Y(y') \, \mathrm{d}P(X, Y) \tag{3}$$

where subscripts emphasize that feature mappings for X and Y may differ, and $\mathcal{C}$ also plays the role of an uncentered covariance operator

$$\mathbb{E}_{XY}[f(X) g(Y)] = \langle f \otimes g, \mathcal{C}_{XY} \rangle \leftrightarrow \langle f, \mathcal{C}_{XY} g \rangle$$

Analogs to variance, $\mathcal{C}_{XX}$, and conditional operators $\mathcal{C}_{Y|X}$ through an embedded version of Bayes' rule follow

$$\mathcal{C}_{Y|X} = \mathcal{C}_{YX} (\mathcal{C}_{XX})^{-1} \tag{4}$$

$$\text{i.e. } \mu_{Y|X} = \mathcal{C}_{YX} (\mathcal{C}_{XX}^{-1}) \phi(x) \tag{5}$$

where the $(\mathcal{C}_{XX})^{-1}$ typically requires some regularization $(\mathcal{C}_{XX} + \epsilon I)^{-1}$ as conditioning on x implies some mapping to a constant beforehand, which may not be formally possible.

Fukumizu et al.[9] provide a proof, as well as a simple analogy. In the case of a Gaussian, the conditional expectation for a linear observable looks like:

$$\mathbb{E}_{Y|X}[a^\intercal Y|X = x] = a^\intercal \Sigma_{YX} \Sigma_{XX}^{-1} x$$

(where Σ are centered covariance operators). Thus, if our embedding is effectively transforming functions and distributions to unique feature space coordinates, then we would expect a corresponding embedded version

$$\mathbb{E}_{Y|X}[g(Y)|X = \cdot] = \left\langle g\Sigma_{YX}\Sigma_{XX}^{-1}, \cdot \right\rangle$$

to give the appropriate conditional expectation.

3.3.2 Practical implementation

Suppose we sample N points x_i in parameter space X – not necessarily following a specific prior distribution – and we denote the joint distribution with resulting observations y_i from Y as $P(X, Y)$. Let $k_X(\cdot,\cdot)$ and $k_Y(\cdot,\cdot)$ be suitable kernels for each space. Through extension, we can incorporate a prior distribution defined by sample weights γ_j and coordinates u_j at each x_i whether or not u_j lie outside the original training set:

$$\hat{m}_\Pi[i] = \sum_j^{N_p} \gamma_j k_X(u_j, x_i)$$

The resulting "kernel mean" reads

$$\hat{\mu} = N\left(G_X + N\epsilon I_N\right)^{-1} \hat{m}_\Pi \tag{6}$$

where G_X is the gram matrix for kernel k_X (over the training data), and ϵ is a regularization factor. By taking $\Lambda = \mathrm{diag}(\hat{\mu})$, and computing the gram matrix G_Y for observations, Fukumizu et al. obtain the matrix

$$R_{X|Y} = \Lambda G_Y \left(\left(\Lambda G_Y\right)^2 + \delta I_N\right)^{-1} \Lambda \tag{7}$$

where δ is another regularization factor, which may be adjusted separately of ϵ.

The posterior for an arbitrary observation $y^\dagger$, and parameter value $x^\star$ then takes the terse form:

$$\mathbf{k}_X^\intercal \underbrace{R_{X|Y}\mathbf{k}_Y(y^\dagger)}_{\rho_i} = k_X(x^\star, x_i)^\intercal R_{X|Y} k_Y(y^\dagger, y_i) \tag{8}$$

where the ρ_i are weights for the kernel contributions of each x_i. The expectation of some function $f(x)$ with respect to the posterior simply reads:

$$f(x_i)^\intercal R_{X|Y} k_Y(y^\dagger, y_i) \tag{9}$$

As the authors remark, many ρ_i may be negative. Though expectations of positive quantities often remain positive, avoiding negative weights is preferable, and necessary for applications like particle filtering. From filtering, we borrow the kernel herding technique to resample while preserving accuracy of expectations (versus crudely clipping or shifting and scaling)[10]. The algorithm is straightforward:

Algorithm 2: Kernel herding

Input: N posterior weights ρ_i and samples x_i from kernel-embedding;
Output: Representative $\bar{x}_i$ resampled with replacement from x_i and given uniform weights;
For initial $j = 1$;

$$\bar{x}_1 \leftarrow \underset{x \in x_1 \ldots x_N}{\mathrm{argmax}} \sum_i^N \rho_i k_X(x, x_i)$$

for $j = 2, \ldots, N$ **do**

$$\bar{x}_j \leftarrow \underset{x \in x_1 \ldots x_N}{\mathrm{argmax}} \sum_i^N \rho_i k_X(x, x_i) - \frac{1}{j}\sum_k^{j-1} k_X(x, \bar{x}_k)$$

end

As the kernel-embedding is data-driven, we discuss a strategy for generating training data here (which may also serve TMCMC initial grids). To help achieve a balance between generality versus excess variability in EEDF representation, we can generate a number of reference EEDFs that bracket (i.e. support) the space of non-zero priors. As a concrete example, we might generate a number of MB and Margenau distributions spanning the expected conditions for a radio-frequency heated plasma setup. Fitting splines (or bases) to these, we can then interpolate and perturb these references with more control than blindly applying MCMC allows. Specifically, we can interpolate reference EEDFs via convex combinations of α_x and/or α_y.

$$\alpha_o \sim \sum_{r \in \text{refs}} w_r \alpha_r \tag{10}$$

where the vector of weights w_r could be drawn from Dirichlet, or other compositional distributions, and involve all, or subsets of the references (if we want to ensure some points land on facets of this effective simplex).

We can also introduce some perturbation via:

$$u\alpha_o + (1-u)\alpha_{\text{pert}};\ u \in [0,1] \tag{11}$$

where α_{pert} can be drawn from a distribution biased to perturb certain parts of the EEDF more aggressively. Of course other schemes are possible.

4 RESULTS AND DISCUSSION

To demonstrate the methods, we used a CR system spanning levels of neutral and singly-ionized Oxygen-16, which included processes of: radiative decay, inelastic and superelastic electron impact transitions, electron impact ionization, and (electron-mediated) three-body recombination. Note though, that enforcing separate density constraints on ionization stages, or any specified subset of levels, is fairly straightforward, and we included this capability. This anticipates incorporating some other knowledge (theoretical, or experimental) to correct for deficiencies in the CR model. For example, if more complex recombination channels involving other species surpass three-body recombination, but are still much slower than most impact transitions, and we have some independent measure of ion populations, then the relative level populations of each ionization stage remain informative.

4.1 Transition Markov Chain Monte Carlo

Again, as TMCMC is more familiar, less convenient by itself, and not our focus here, we just give a couple of illustrative examples. The examples shown employed a uniform prior.

First, we obtained satisfactory results when some element(s) in the initial grid was in the ballpark of the target. However, the typical multivariate normal proposal distribution did not fare well in this context, but a uniform box proposal informed by neighboring points did (remember that the approach relaxes typical MCMC constraints on proposal adaptivity).

Fig.3 shows results with the method using four percentile bins, and exponential-tension splines (in linear space), which manages to imitate MB-like distributions at comparable degrees of freedom. Increasing degrees of freedom permits increasingly oscillatory solutions

Fig.4 shows results for a run that used 7 M-splines of order 3 with free relative weights and internal knots (again in linear space). Note that the first M-spline with finite value at the origin is excluded from this representation to automatically preserve the boundary condition. The weights and intervals between knots were related to α_x, α_y, and corresponding alogr as mentioned in §3.1. This case had a total of 10 degrees of freedom, and the initial particle net was a regular lattice in alogr space, which encompassed a wider variety of EEDF shapes than the previous example. Even with a moderate number of well-chosen bases functions, this example illustrates how some artifacts of the bases can appear (specifically the reverse-knee around 20-40eV for some of the lower

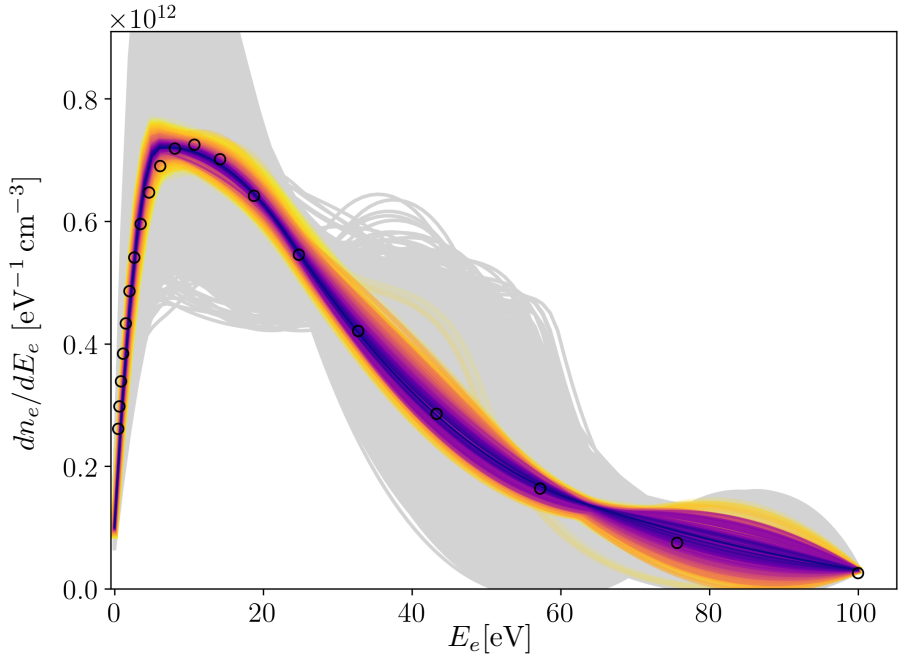

Figure 3: An example TMCMC run using exponential tension-splines, four fixed percentile partitions (3 degrees of freedom). EEDFs with higher posterior weight are shown as thinner, darker, foreground curves; less likely EEDFs are shown fainter, lighter, and further back. The background of gray curves shows the initial grid, and black circles show the EEDF used to generate target observables.

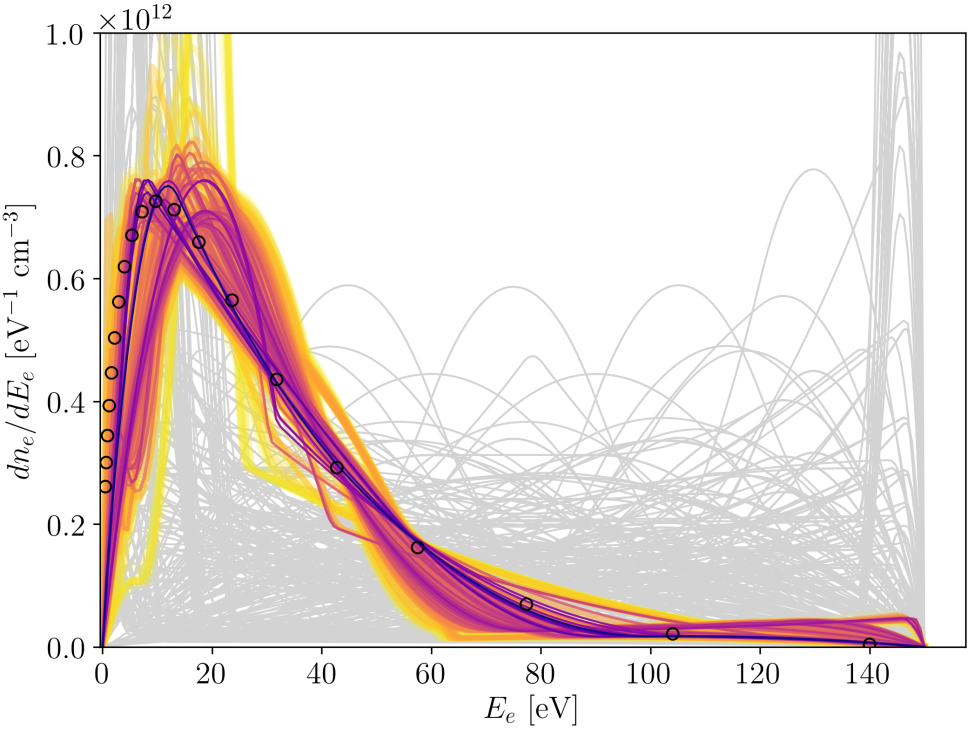

Figure 4: An example TMCMC run using I-splines as (CDF) basis functions, with the same plotting conventions as fig3

posterior curves as the method manages to match the observations by shifting the last spline leftward instead of reducing its weight).

This subset of the experiments with representations sketches how the parameterization with the TMCMC method formally meets our objectives, but a more efficient alternative is desirable.

4.2 Kernel-embedded Posterior

To demonstrate application of the KE method, we generate training data mimicking an MB EEDF subject to inelastic transitions by adding shifted copies of the MB, as well as random perturbations according to eqn.10 – see fig.5. For these examples we used the representation shown in fig.2: exponential tension splines in log-log space.

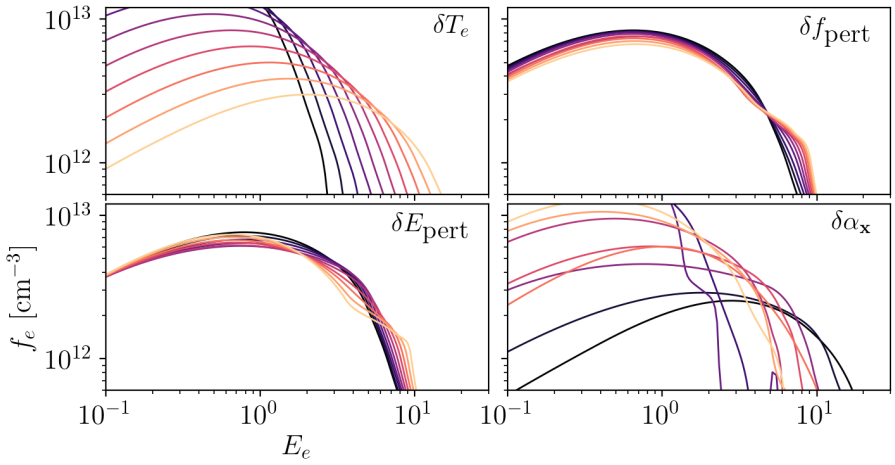

Figure 5: Top left panel shows 'spine' of MB distributions; lower left one of the curves with a fixed fraction echoed at higher energies; upper right varying fractions echoed at a fixed energy; lower right shows some of the random perturbations.

Fig.6 shows the training EEDFs with highest ρ_i (eqn.8) calculated for a test EEDF given uniform prior over the training data, and 7 observed level populations. The latter are plotted below (fig.7) in their rescaled form

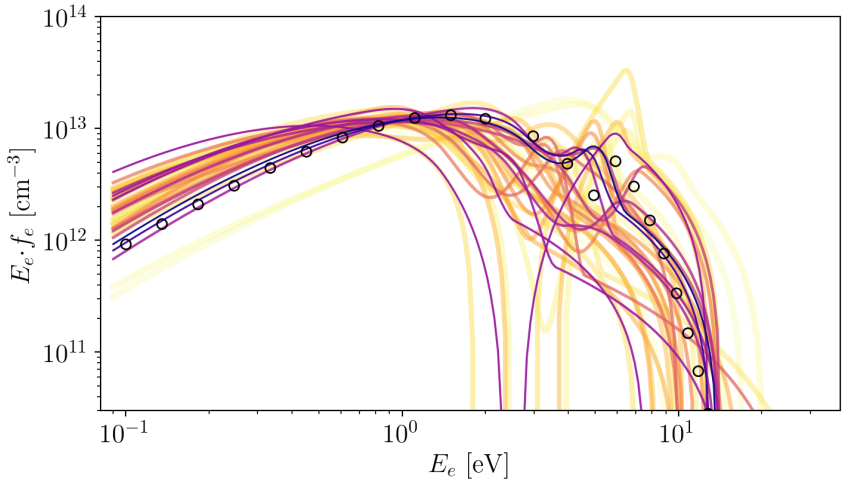

Figure 6: Scheme similar to fig.3 except that the darker curves correspond to higher raw ρ_i

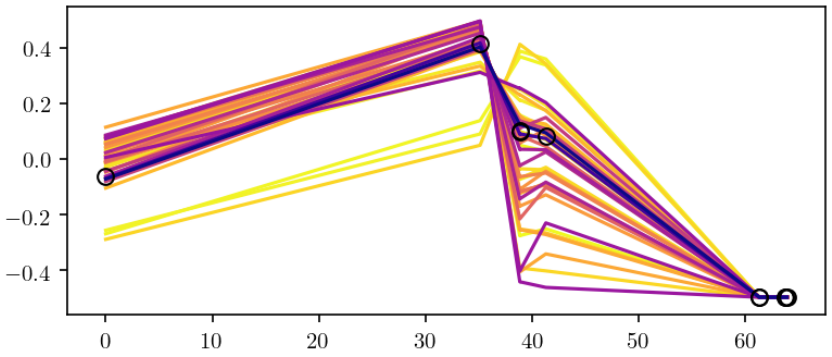

Figure 7: Transformed level populations plotted in similar fashion to the EEDFs. Plotting as curves versus points is merely to improve visibility and highlight crossings.

Recall that the ρ_i may fluctuate, sometimes enough to pull binned quantities significantly below zero as fig.8 shows for alogr values weighted by ρ_i. The same figure shows a corresponding histogram for out-of-sample parameter values near high posterior points from the training data, after kernel herding (alg.2) has been applied.

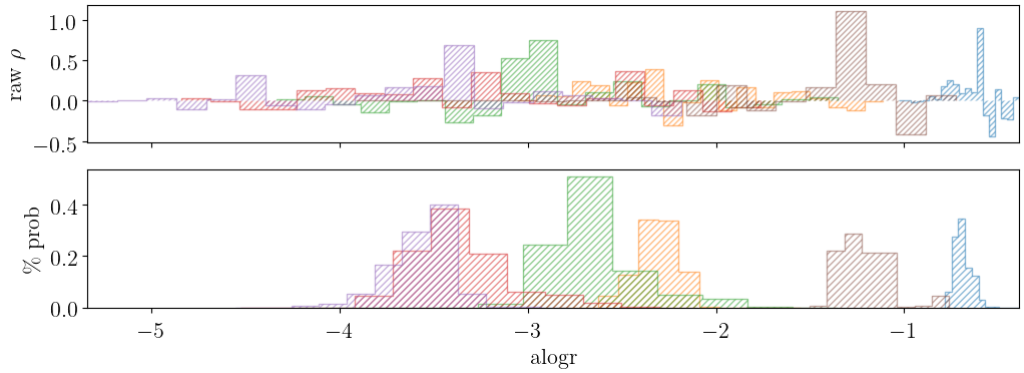

Figure 8: Top panel shows histograms for the alogr_i coordinates weighted by raw ρ_i values. The bottom panels shows the resulting positive distribution, at alogr_i extended beyond the training data, by using kernel-herding.

Fig.9 shows these out-of-sample EEDFs with posterior weights using kernel-herding. The method for sampling new EEDFs led to some slight clustering, but overall the solution looks surperior, which may seem strange given the appearance of fig.6, and that we have not calculated new obserations for these EEDFs. The moral is that the joint distribution was 'learned' fairly accurately, and that the raw ρ_i should be interpreted carefully. Applying a strong MB prior, which distributes 90% total weight to the MB curves (upper left of fig.5), and 10% to all the rest emphasizes this lesson. See figures 10–12.

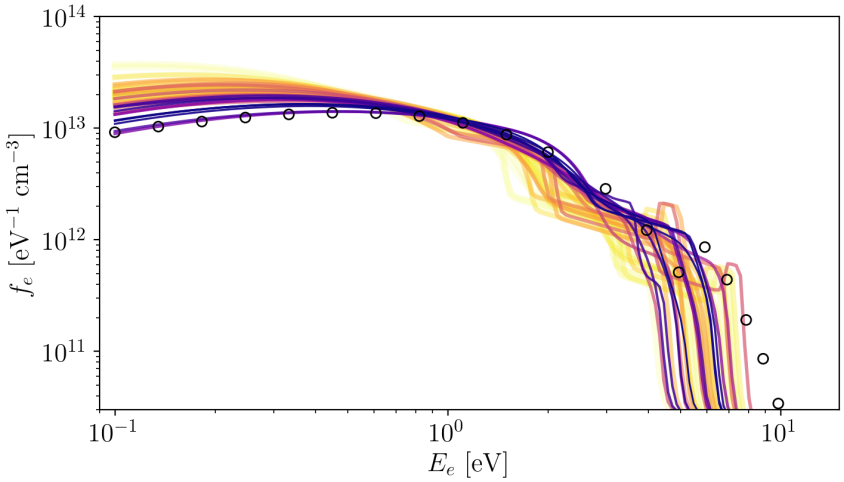

Figure 9: Similar to fig.6 except that the EEDFs shown are not in the original training data, and the weights have been adjusted through kernel herding. For more clarity, only every fourth EEDF in the extended parameters, ranked by posterior, is shown.

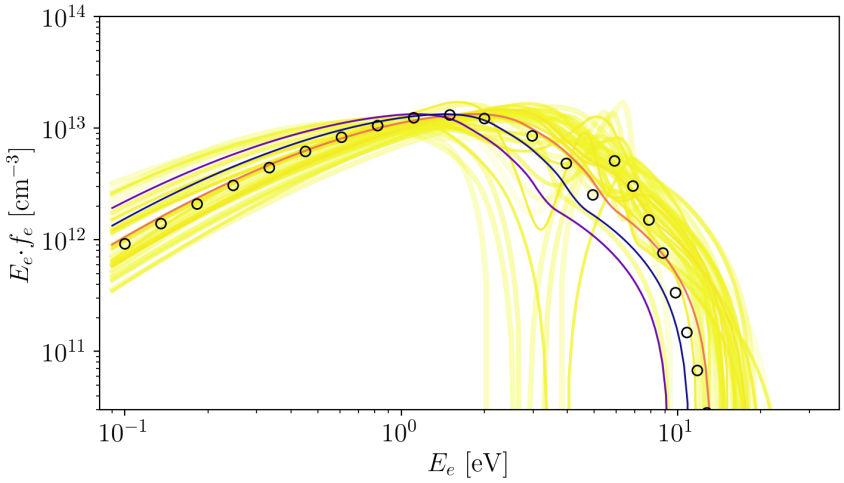

Figure 10: Similar to fig.6 now for a strong MB prior. Note how the "closest" MB curve has lower ρ_i, but many more non-MB "neighbors" with enough weight to meet the plotting threshold.

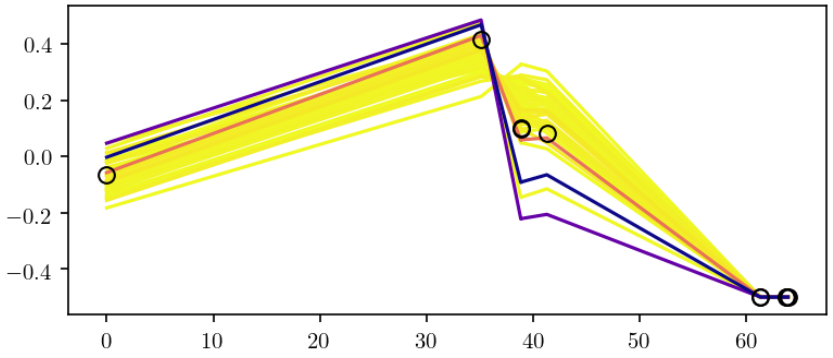

Figure 11: For completeness, the corresponding plot to fig.7, which shows the same effect as fig.11.

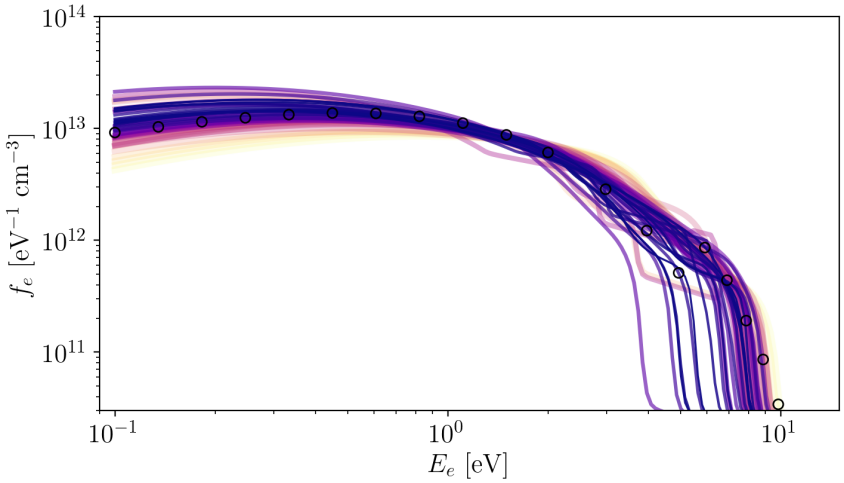

Figure 12: Similar to fig.9, but for the MB prior. The "winning" curves are all, unsurprisingly, much more MB in shape.

5 RECOMMENDATIONS

The most urgent next step is practical application (furthering motivation for a report illustrating the basic concept).

Extension to time-varying EEDF inference was also intended from the inception of this work. Mirroring the situation here, we developed an approach using a more conventional sequential Monte Carlo (SMC, e.g. [11]) for another effort focused on static parameter inference given a known (or accurately modeled) EEDF. The method lends itself to state estimation as well, but if our focus is solely on state estimation of the time-dependent EEDF, then we can also try extant kernel-embedding approaches here (e.g. the paper referenced for kernel-herding[10]).

Extension to anisotropy requires some additional constraints or diagnostic channels like line shifts. The specific CDF parameterization will no longer apply, though the compositional coordinate techniques should still find use. The overall KE framework generalizes to anisotropy, and many other related inference problems. For example, even in situations where local thermal equilibrium holds, KE can help tackle complex spatial dependence.

Further developing the computational machinery (e.g. intrinsically reducing fluctations, more careful bandwidth assignment, faster Gram matrix evaluations...) is obviously important, and some aspects have already been pursued. For generic EEDF inference however, since implementation of early kernel embedding methods already lends itself to realistic tests, yet practical demonstration is lagging, the latter should arguably take priority over optimizing the method.

References

[1] J. S. Bear, "A Logistic Normal Mixture Model for Compositions with Essential Zeros," 2016.

[2] B. J. McCartin and A. Jameson, "Numerical solution of nonlinear hyperbolic conservation laws using exponential splines," *Computational mechanics* **6**(2), pp. 77–91, 1990.

[3] B. J. McCartin, "Theory of exponential splines," *Journal of Approximation Theory* **66**(1), pp. 1–23, 1991.

[4] J. Ching and Y.-C. Chen, "Transitional markov chain monte carlo method for bayesian model updating, model class selection, and model averaging," *Journal of engineering mechanics* **133**(7), pp. 816–832, 2007.

[5] W. Betz, I. Papaioannou, and D. Straub, "Transitional markov chain monte carlo: observations and improvements," *Journal of Engineering Mechanics* **142**(5), p. 04016016, 2016.

[6] M. Khalil, J. Lao, C. Safta, and H. N. Najm, "Transitional markov chain monte carlo sampler in uqtk.," 3 2020.

[7] L. Song, K. Fukumizu, and A. Gretton, "Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models," *IEEE Signal Processing Magazine* **30**(4), pp. 98–111, 2013.

[8] K. Fukumizu, L. Song, and A. Gretton, "Kernel bayes' rule: Bayesian inference with positive definite kernels," *The Journal of Machine Learning Research* **14**(1), pp. 3753–3783, 2013.

[9] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces," *J. Mach. Learn. Res.* **5**, pp. 73–99, Dec. 2004.

[10] M. Kanagawa, Y. Nishiyama, A. Gretton, and K. Fukumizu, "Filtering with state-observation examples via kernel monte carlo filter," *Neural computation* **28**(2), pp. 382–444, 2016.

[11] N. Chopin, P. E. Jacob, and O. Papaspiliopoulos, "SMC2: An Efficient Algorithm For Sequential Analysis Of State Space Models," *J. Royal Statistical Society B* **75**(3), pp. 397–426, 2013.

6 LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

CDF	cumulative distribution function
CR	collisional-radiative
EEDF	electron energy distribution function
KE	Kernel-embedding
MB	Maxwell-Boltzmann
MCMC	Markov chain Monte Carlo
TMCMC	Transitional Markov chain Monte Carlo