Transforming Worlds: Automated Involutive MCMC for Open Universe Probabilistic Models

George Matheos[*,1,2] Alexander K. Lew [*,1] Matin Ghavamizadeh[1,2] Stuart Russell[2] Marco Cusumano-Towner[1] Vikash K. Mansinghka[1]

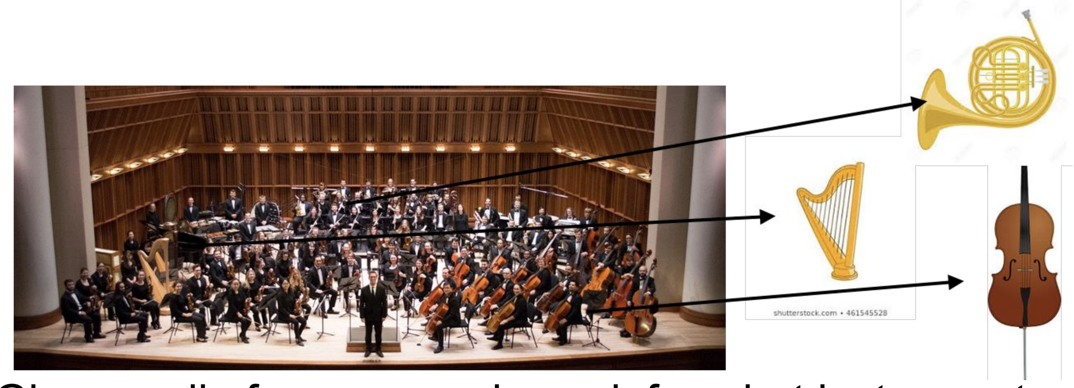
[*] Equal Contribution [1]MIT Probabilistic Computing Project [2]UC Berkeley

Syndicated Submission. Please cite *Matheos, Lew et al. "Transforming Worlds…". Advances in Approximate Bayesian Inference 2021.*

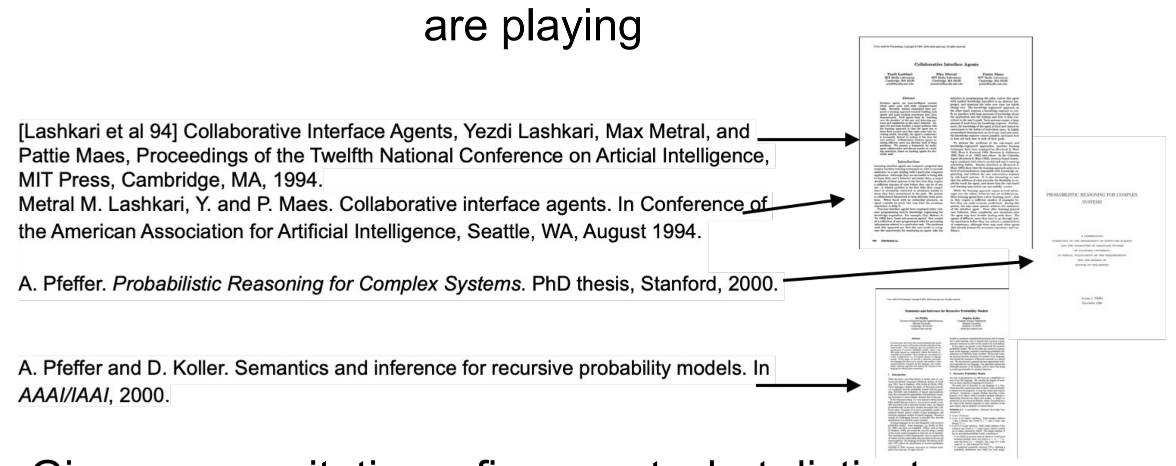
1. Overview: Inference in Open Universe Models

An *open universe probabilistic model* (OUPM) describes uncertainty in **how many** objects exist, as well as in their relationships and properties.

Example OUPM Inference Problems



Given audio from a symphony, infer what instruments are playing



Given paper citations, figure out what distinct papers they refer to

Systems for Inference in OUPMs

	Applicable to OUPMs?	*Broad Inference Kernel Class?**	*Math automated?*	*Automated Incremental Computation for OUPMs?*	*User Must Write*
BBVI, HMC, …	No	N/A	N/A	N/A	N/A
Lightweight MH (BLOG)	Yes	No	Yes	Yes	Nothing
Manual RJ-MCMC (BLOG)	Yes	Yes	No	No	Java Code to compute acceptance probability and transform low-level data structures
Stochaskell RJ-MCMC	Yes	Intermediate	Yes	No**	Automatically -reversible Haskell Code to transform a trace
Automated Involutive MCMC (Gen)	Yes	Yes	Yes	No	Code in low-level MCMC kernel DSL to transform a PPL trace and specify a reverse move
Automated Involutive MCMC for OUPMs (Ours)	**Yes**	**Yes**	**Yes**	**Yes**	**Code in new OUPM Inference DSL to stochastically modify a high-level "world" and specify reverse move**

Our Contributions

- **An MCMC kernel DSL for transforming open-universe "worlds" with high-level syntax**
- **Algorithms to efficiently and automatically implement Involutive MCMC for OUPMs from high-level specs; proofs of correctness**
- **A new formalism for OUPMs with continuous variables**

2. New DSLs in Gen for Open-Universe Modeling and Inference

We introduce a new DSL for writing open-universe models in Gen, and a new DSL for writing inference kernels for them.

The model probabilistic program defines a distribution over "possible worlds" of interrelated objects.

The inference kernel is a probabilistic program which outputs (1) a *world update specification*, and (2) a *reverse move specification*.

Such a program defines an MCMC kernel in the class of *involutive MCMC kernels* (Cusumano-Towner et al 2020.). **Our system automates the efficient implementation of the kernel**.

Model Code

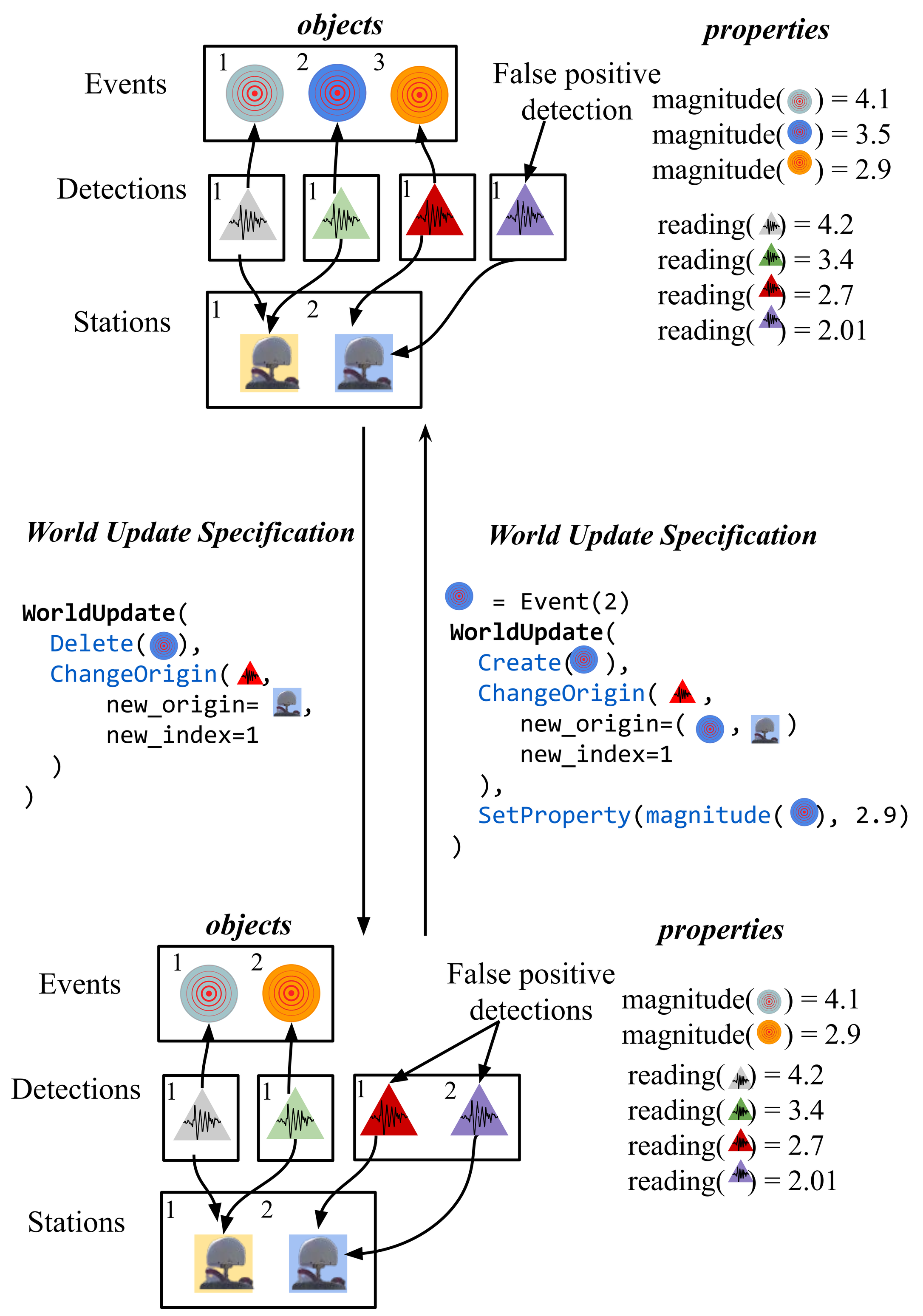
```
type Event
number Event() ~ poisson(5)
type Station
number Station() = 2
type Detection
number Detection(::Station, ::Event) ~ bern(0.8)
number Detection(::Station) ~ poisson(1)

property magnitude(::Event) ~ exponential(3.0)
property function reading(d::Detection)
   if is_false_positive(d)
       return reading ~ normal(2.0, 0.5)
   else
       (station, event) = origin(d)
       event_mag = get(magnitude(event))
       return reading ~ normal(event_mag, 0.2)
   end
end
function is_false_positive(d::Detection)
   return length(origin(d)) == 1
end
observation_model function detections()
   detections = get_object_set(Detection)
   return [get(reading(d)) for d in detections]
end
```

Seismic monitoring model inspired by **Arora et al.** "Net-Visa…".

Inference Kernel Code

```
kernel function birth_death_kernel(prev_world)
 # ... compute p_birth, false_positive_dets, events ...
 do_birth_move ~ bern(p_birth)
 if do_birth_move
   detection_of_new_event ~ uniform(false_positive_dets)
   new_evt_mag ~ normal(
        get(reading(detection_of_new_event)), 0.2)
   new_evt_idx ~ uniform(range(1, length(events) + 1))
   new_evt = Event(new_evt_idx)
   return
     WorldUpdate(
       Create(new_evt),
       ChangeOrigin(detection_of_new_event,
          new_origin=(station, new_evt), new_index=1),
       SetProperty(magnitude(new_evt), new_evt_mag)
     ),
     ReverseMove(
       :do_birth_move  => false, :evt_to_delete => new_evt,
       :fp_det_idx => index(fp_det)
     )
 else
   # ... compute dubious_events, num_fps_at_station ...
   evt_to_delete ~ uniform(dubious_events)
   fp_det_idx ~ uniform(range(1, num_fps_at_station + 1))
   station = origin(detection_of_evt_to_delete)[1]
   return WorldUpdate(...),  ReverseMove(...)
 end
end
```

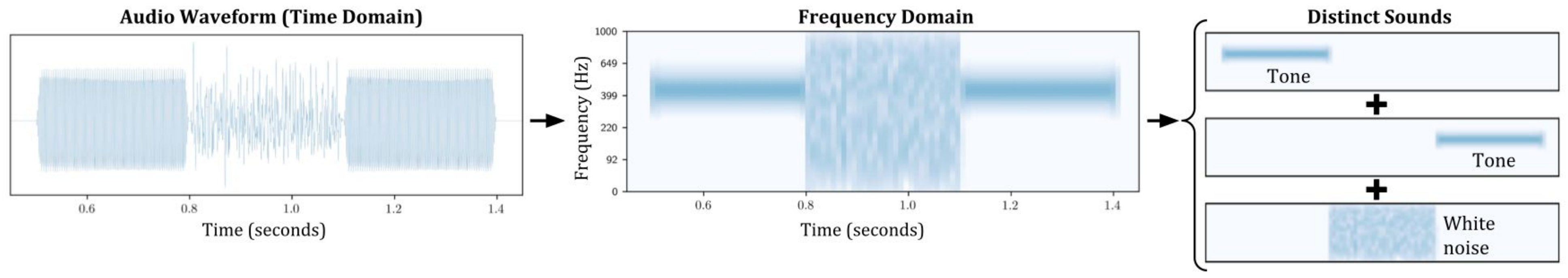
Example worlds before/after update

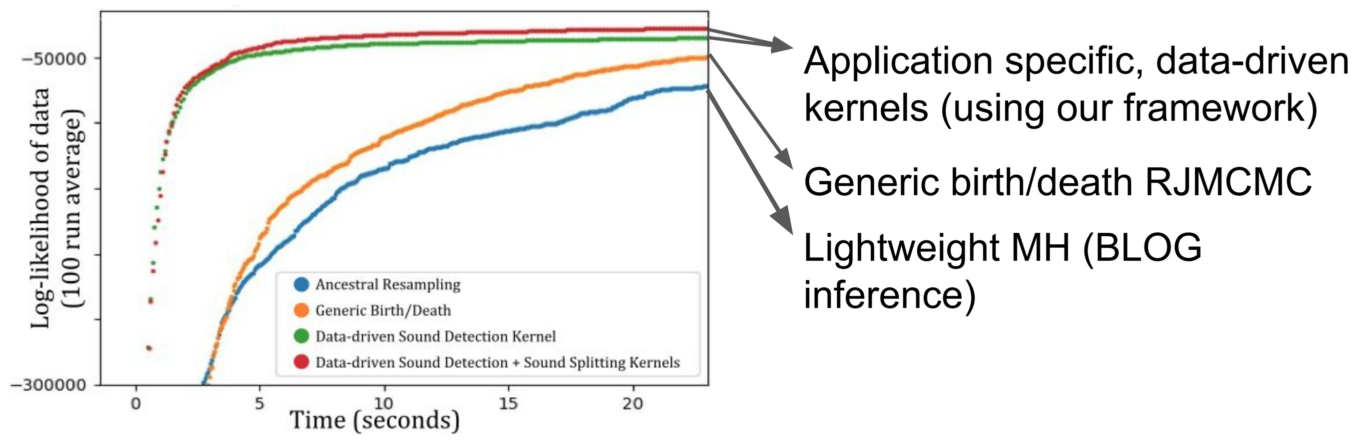


3. Other Examples

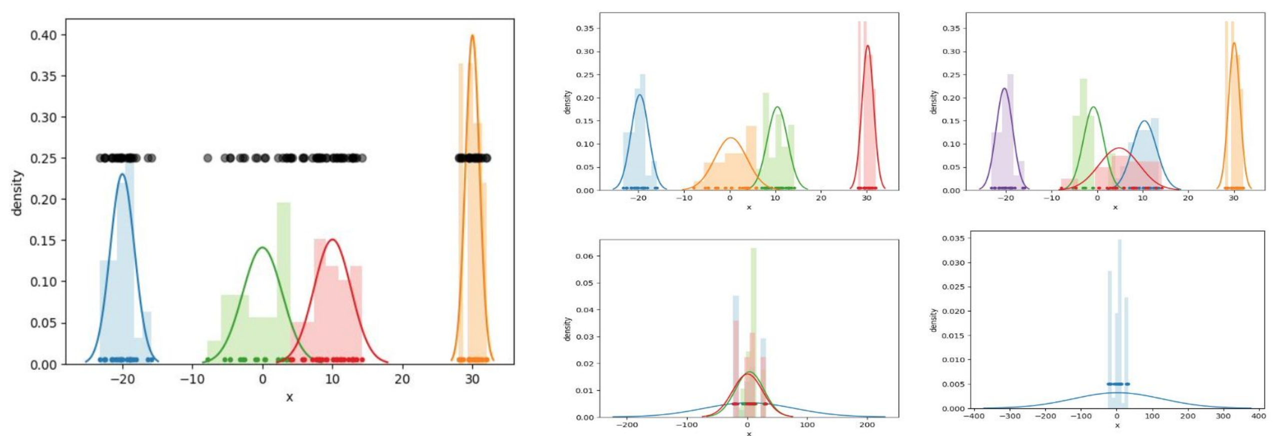
Custom, data-driven MCMC outperforms generic MCMC. Our system automates the math and efficient implementation of custom MCMC from high-level transition kernel descriptions ("world transformations").

Inferring Audio Sources from an Audio File (Cusimano et al. 2018)





Mixture Model w/ Unknown Number of Components (Richardson & Green 1997)



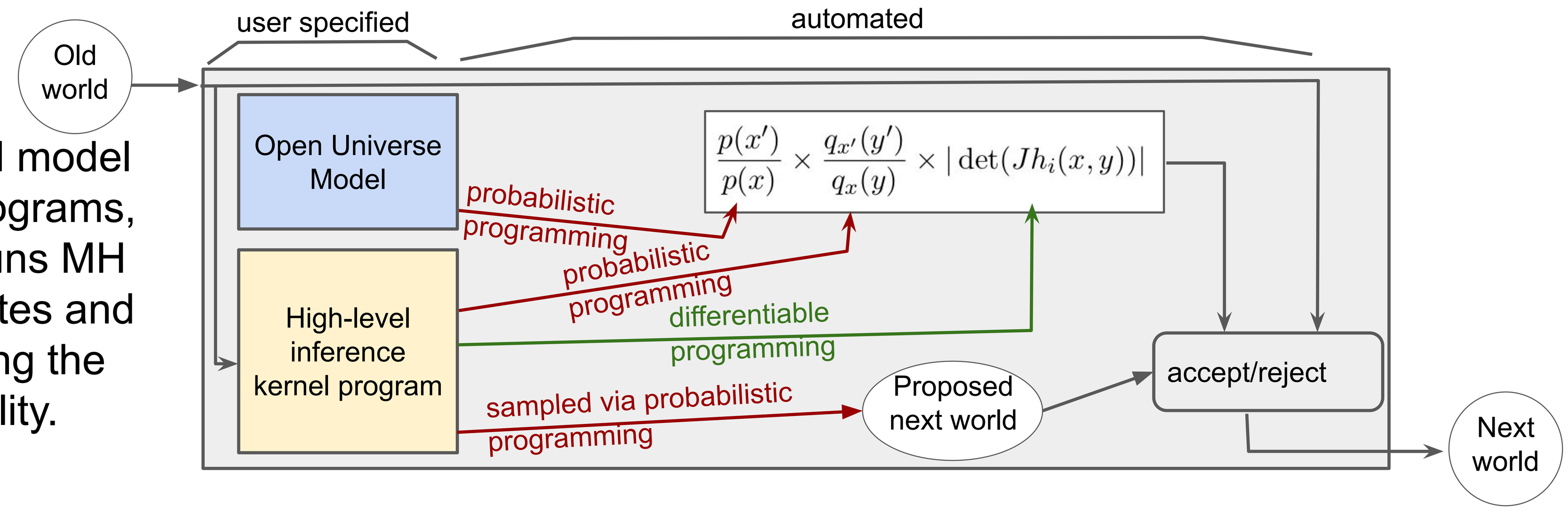
Richardson & Green's Smart Inference Kernels (using our framework)

Lightweight Metropolis Hastings

Ground truth clusters | Inferred clusters

4. Automating Inference Kernel Implementations

Using the user-provided model and inference kernel programs, our system efficiently runs MH by sampling world updates and automatically computing the acceptance probability.



See paper for automation algorithm details (**Algorithm 1**; **Algorithm 3**).

5. Next Steps

- Support custom, data-driven SMC inference (as well as MCMC inference)
  - Build on preliminary research on new "involutive SMC" framework in Gen.
- Work toward effective, automated SMC + MCMC inference algorithms for restricted classes of OUPMs (perhaps parametrized by user-provided object detectors)
- Use techniques from inference amortization to optimize parameters in proposal distributions
- Improve inference program wall-clock performance via compilation.

6. References

Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, Daniel L. Ong, and Andrey Kolobov. BLOG: Probabilistic models with unknown objects. In Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI 2005, pages 1352–1359. Morgan Kaufmann Publishers Inc., **2005a**.

Marco Cusumano-Towner, Alexander K. Lew, and Vikash K. Mansinghka. Automating involutive mcmc using probabilistic and differentiable programming. 2020.

David A. Roberts, Marcus Gallagher, and Thomas Taimre. Reversible jump probabilistic programming. volume 89 of Proceedings of Machine Learning Research, pages 634–643. PMLR, 16–18 **Apr 2019**. URL http://proceedings.mlr.press/v89/roberts19a.html.

Maddie Cusimano, Luke B Hewitt, Josh Tenenbaum, and Josh H McDermott. Auditory scene analysis as bayesian inference in sound source models. In CogSci, **2018**.

Sylvia. Richardson and Peter J. Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). Journal of the Royal Statistical Society: Series B (Statistical Methodology), 59(4):731–792, **1997**. doi: 10.1111/1467-9868.00095.

Arora, Nimar S., Stuart Russell, and Erik Sudderth. "NET-VISA: Network processing vertically integrated seismic analysis." *Bulletin of the Seismological Society of America*103.2A (**2013**): 709-729.