Policy Learning for Continuous Space Security Games using Neural Networks

Nitin Kamra[1], Umang Gupta[1], **Fei Fang**[2], Yan Liu[1], Milind Tambe[1]
University of Southern California[1], Carnegie Mellon University[2]
nkamra, umanggup, yanliu.cs, tambe@usc.edu[1],
feifang@cmu.edu[2]

Stackelberg Security Game (SSG)

- A leader-follower game with broad applications

Physical Infrastructure

Transportation Networks

Cyber Systems

Environmental Resources

Endangered Wildlife

Fisheries

Stackelberg Security Game (SSG)

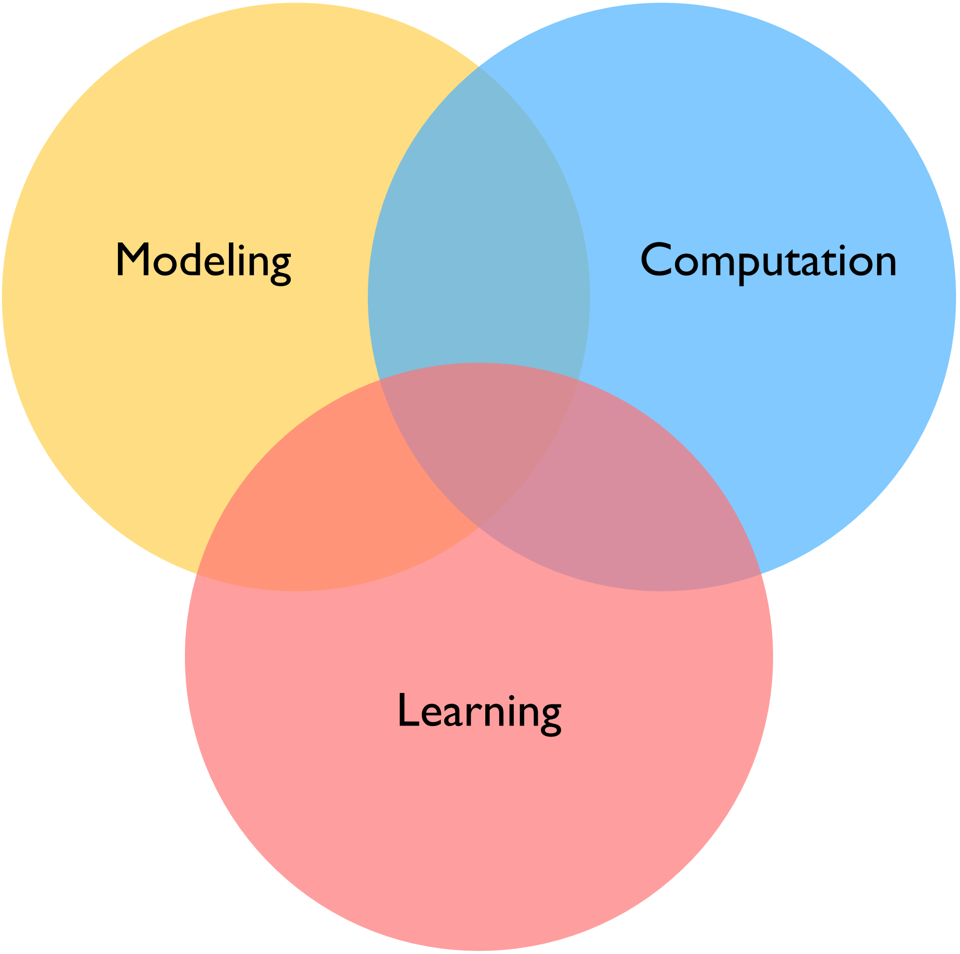
- A leader-follower game with broad applications

- Basic model:
  - Defender allocate limited resources to protect targets
  - Attacker choose a target to attack after surveillance
  - Goal: Find optimal defender strategy

Adversary

Defender			Target #1	Target #2
	55.6%	**Target #1**	5, -3	-1, 1
	44.4%	**Target #2**	-5, 4	2, -1

Research Efforts in SSGs

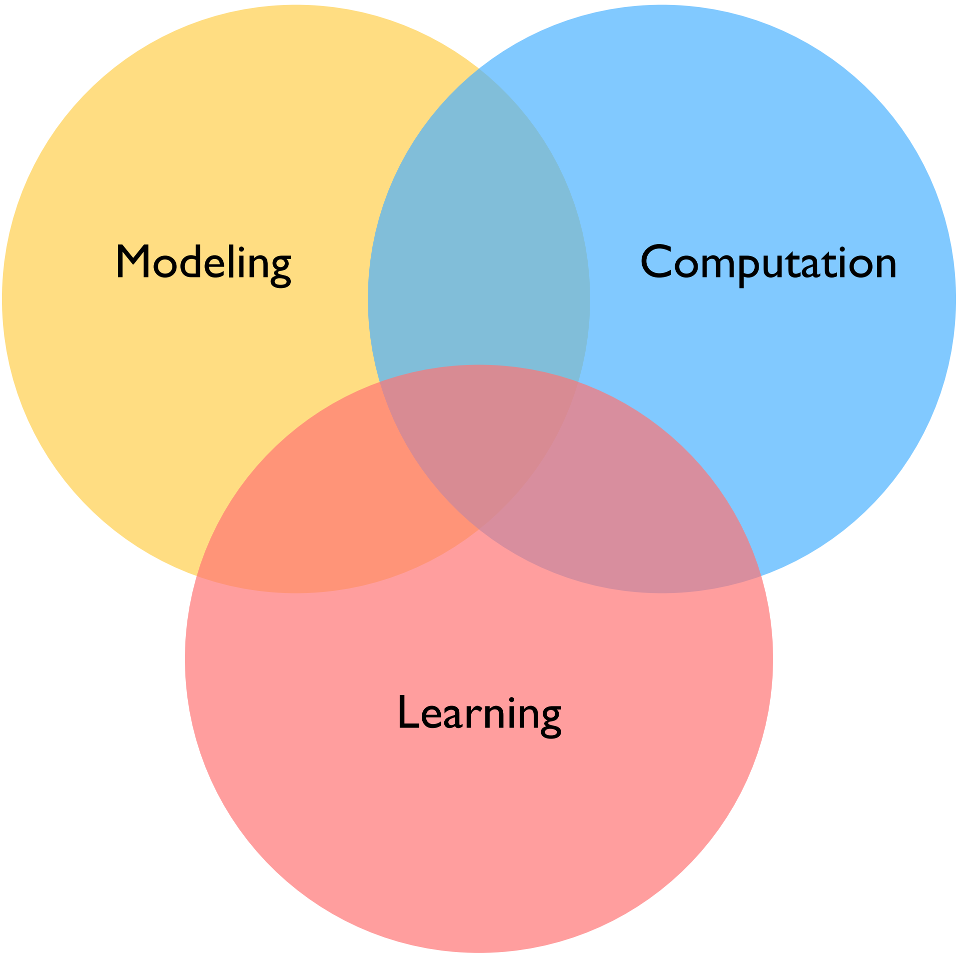


Research Efforts in SSGs

- Model and address complex real world problems
  - Continuous space/time
    - Fang et al., 2013; Gan et al., 2017
  - Repeated/Sequential/Dynamic interaction
    - Fang et al., 2015; Lisy et al., 2016
  - Information
    - Durkota et al., 2015; Xu et al., 2018

- Solution approaches for continuous space/time
  - Discretization
    - Fang et al., 2016
  - Exploit special spatio-temporal structure, e.g., symmetric circular shaped forest
    - Johnson et al., 2012

Research Efforts in SSGs

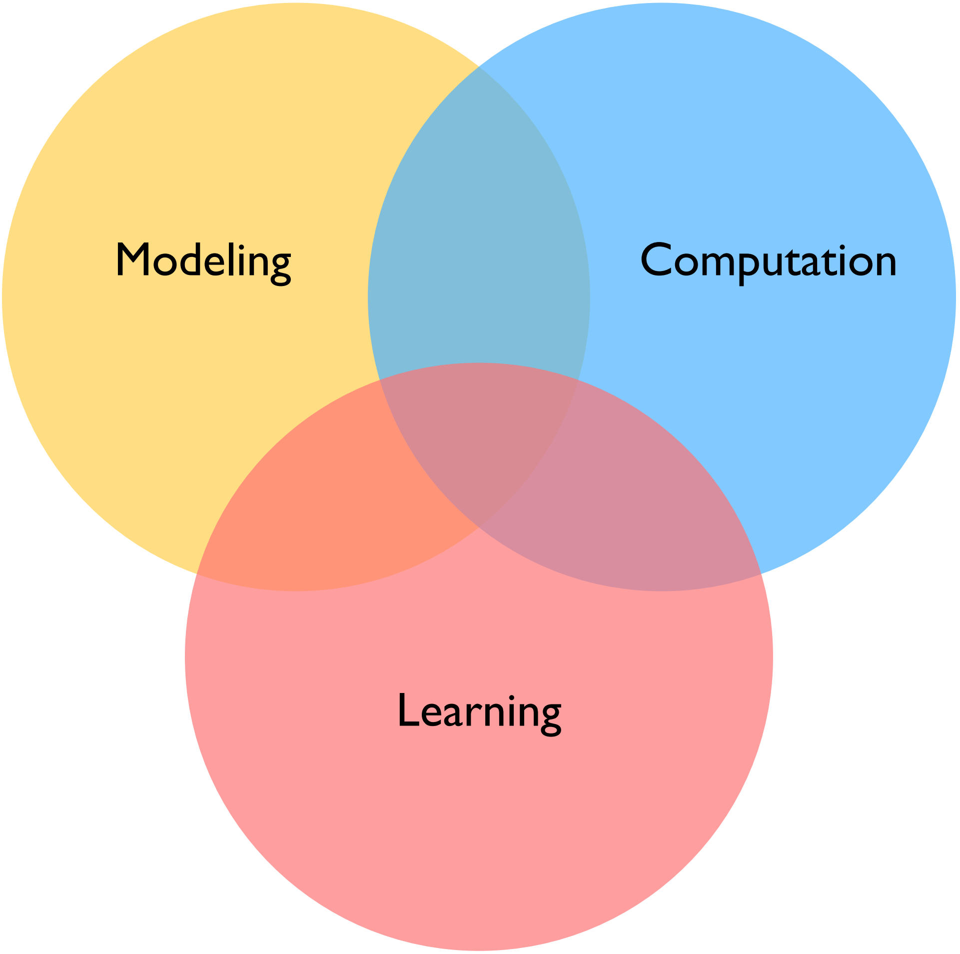


Research Efforts in SSGs

- Compute optimal defender strategy
  - Scaling up
    - Bosanský et al., 2015; Kiekintveld et al., 2009; Basilico et al., 2012
  - Uncertainty & Robustness
    - Haskell et al., 2014; Jiang et al., 2013; Nguyen et al., 2015; Bo et al., 2011

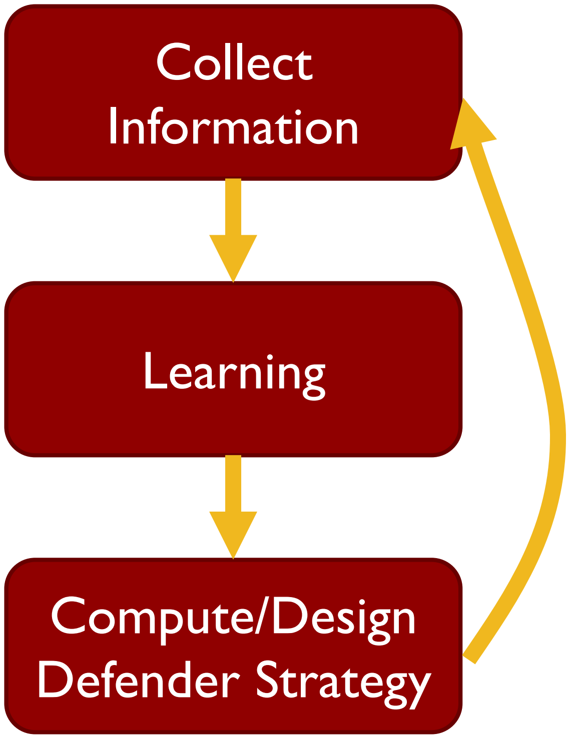
- Solution approaches for scaling up
  - Mathematical Programming based approaches
    - Conitzer & Sandholm, 2006; Paruchuri et al., 2008; Jain et al., 2011
  - Abstraction
    - Basak et al., 2016
  - Gradient descent
    - Amin et al., 2016

Research Efforts in SSGs

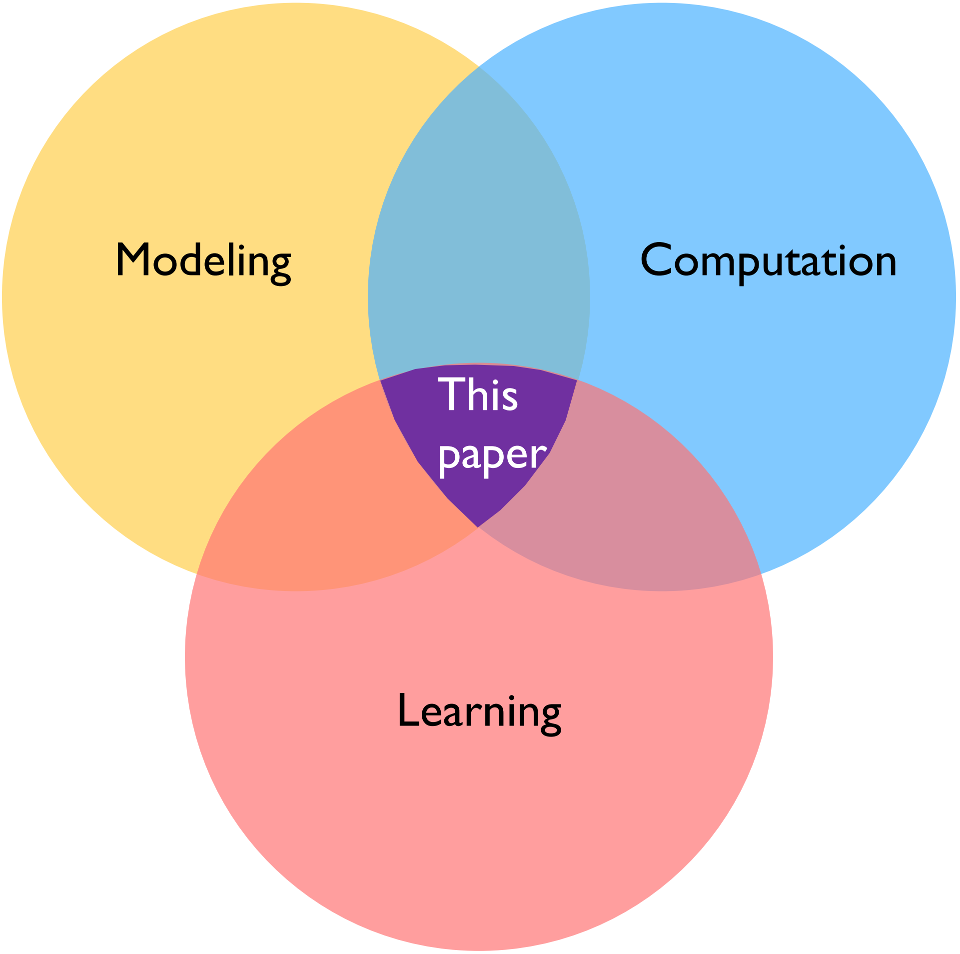


Research Efforts in SSGs

- Learn key elements in games
  - Payoff
    - Blum et al., 2014; Balcan et al., 2015
  - Opponent behavior
    - Yang et al., 2014; Kar et al., 2016; Nguyen et al., 2016; Sinha et al., 2016; Haghtalab et al., 2016



Policy Learning for Continuous Space Security Games



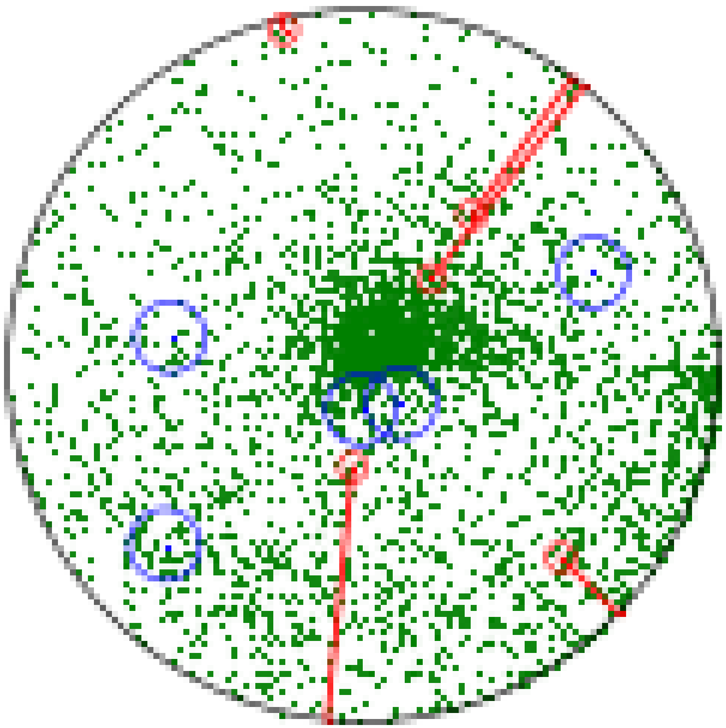
Policy Learning for Continuous Space Security Games

- This paper: Compute optimal defender policy through policy learning from self play

- Contributions
  - A new way of handling continuous space security games
  - Augment existing toolbox for computing optimal strategy
  - Learn a "policy": mapping from game elements to strategy

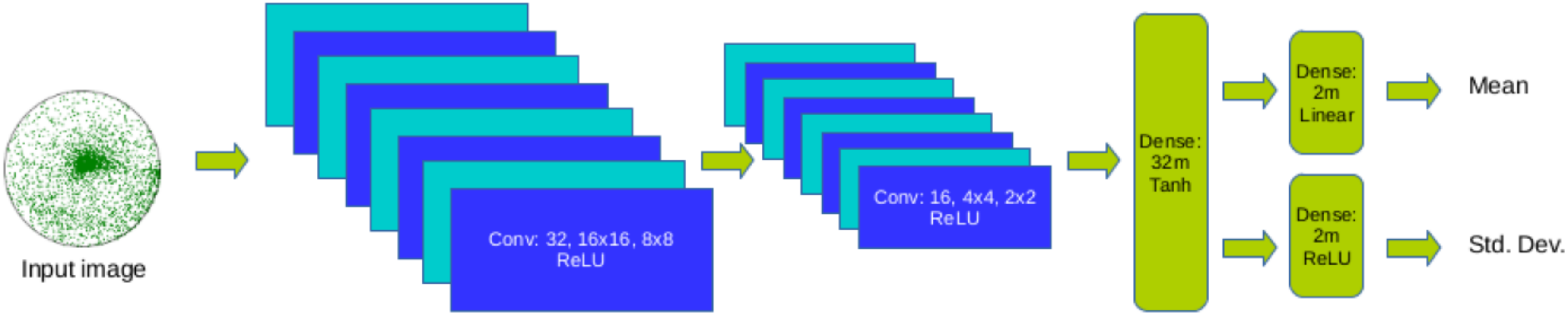
Policy Learning for Continuous Space Security Games

- Green dots: Valuable trees
- Blue circles: Defender location
- Red circles: Logging locations
- Goal: Find defender strategy or defender policy
  - Tree distribution → defender strategy



Policy Learning for Continuous Space Security Games

- Represent defender policy with CNN
  - Image→Mean/Std of radius and angle (→Guard location)
- Attacker's policy represented in a similar way



Policy Learning for Continuous Space Security Games

Algorithm 1: OptGradFP

Initialization. Initialize policy parameters $\boldsymbol{w_D}$ and $\boldsymbol{w_O}$, replay memory *mem*;

for *ep in* $\{0, \ldots, ep_{max}\}$ **do**

Simulate n_s game play. Sample game setting and actions from current policy π_D and π_O n_s times, save in *mem*;

Replay for defender. Draw n_b samples from *mem*, resample defender action from current policy π_D;

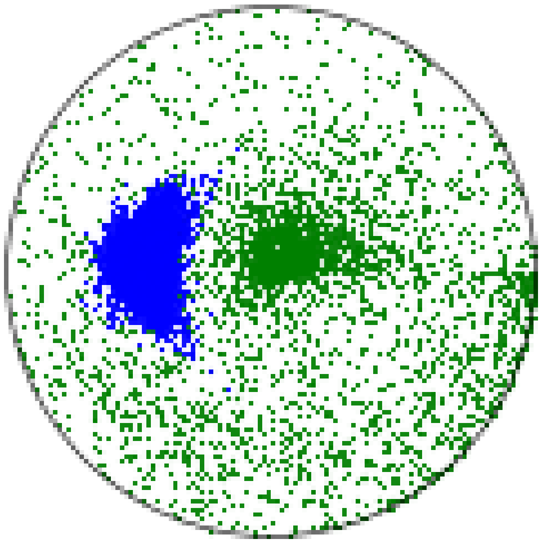
Update parameter for defender. Update defender policy parameter $\boldsymbol{w_D} := \boldsymbol{w_D} + \frac{\alpha_D}{1+ep\,\beta_D} * \nabla_{\boldsymbol{w_D}} J_D$;

Replay for attacker. Draw n_b samples from *mem*, resample attacker action from current policy π_O;

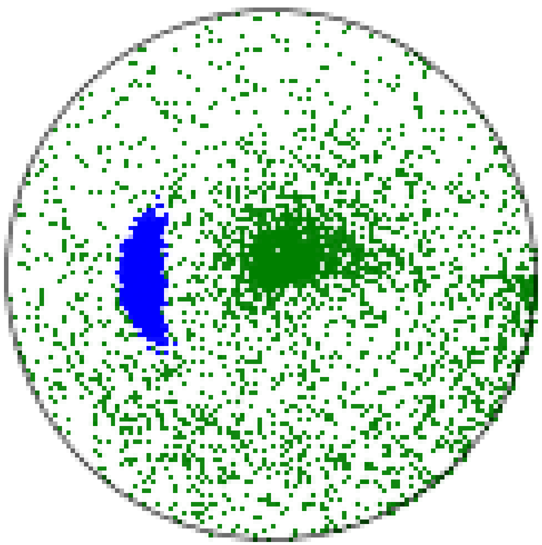
Update parameter for attacker. Update attacker policy parameter $\boldsymbol{w_O} := \boldsymbol{w_O} + \frac{\alpha_O}{1+ep\,\beta_O} * \nabla_{\boldsymbol{w_O}} J_O$

Policy Learning for Continuous Space Security Games

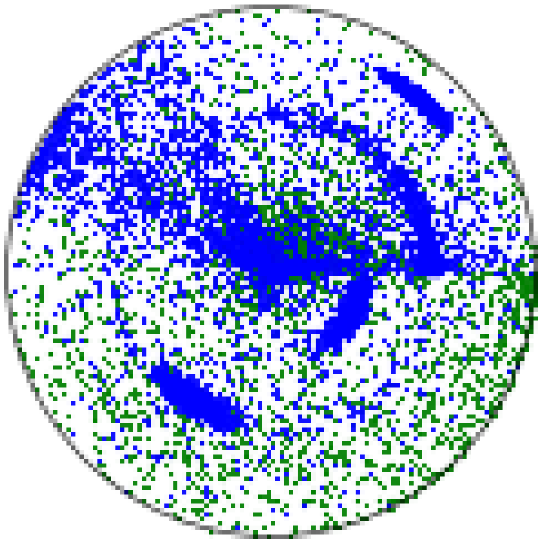
- Single game state



Cournot Adjustment



StackGrad



OptGradFP

- Multiple game state
  - Train on 1000 forest states, predict on unseen forest state
  - 7 days for training, Prediction time 90 ms

Summary

- Policy Learning for Continuous Space Security Games using Neural Networks
  - No discretization
  - Policy learning + Fictitious play + Deep learning
  - Shift computation from online to offline

Thank you!

Fei Fang
feifang@cmu.edu

This research was supported in part by NSF Research Grant IIS-1254206, MURI grant W911NF-11-1-0332, Harvard Center for Research on Computation and Society fellowship and USC Viterbi Graduate PhD fellowship.