Adversarial Training for Textual Entailment with Knowledge-Guided Examples

jcykcai

Motivations

- Datasets tend to be homogeneous.
- Models overfit to repetitive patterns, but fail to cover long-tail patterns or linguistic phenomena such as negation.

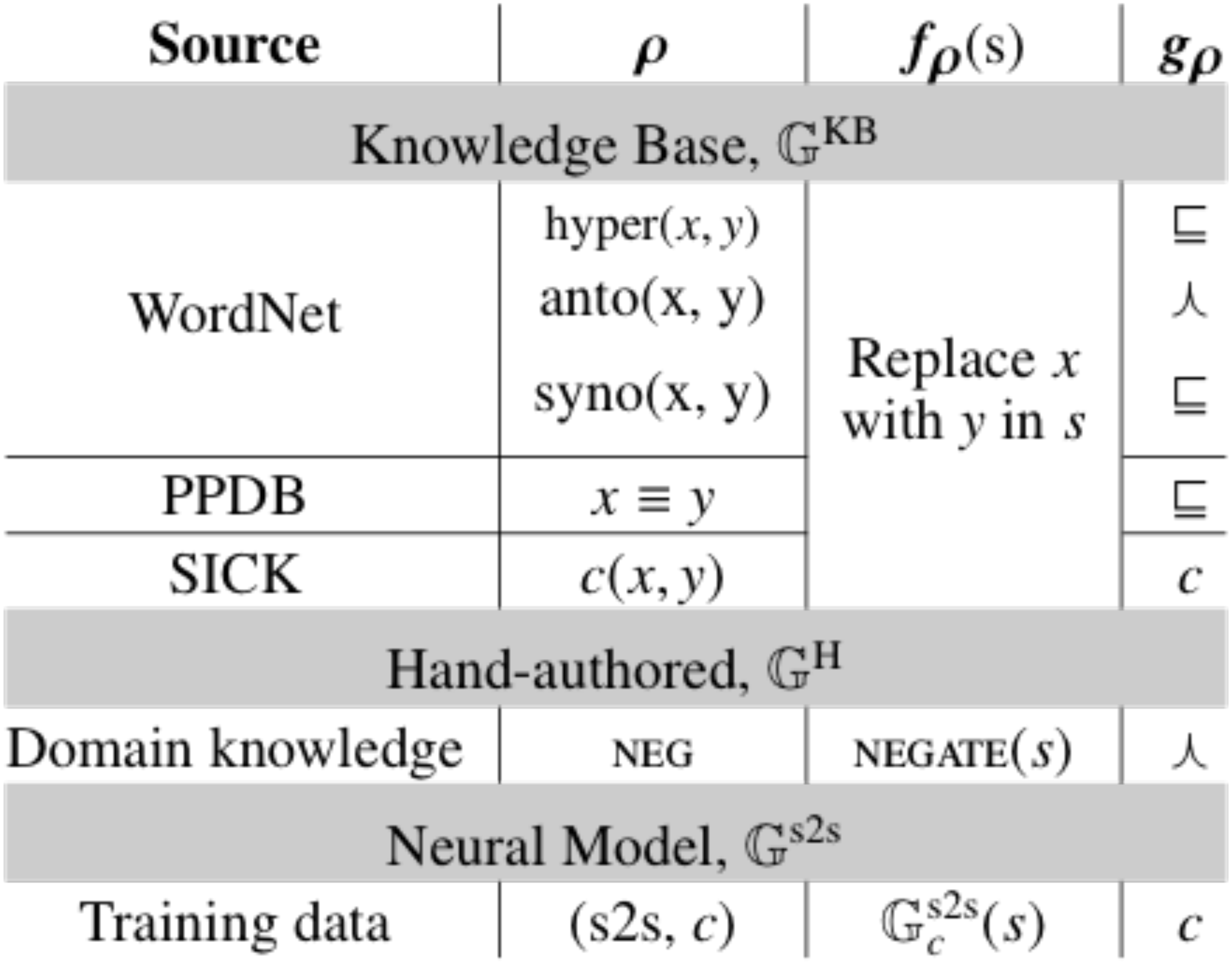
Motivations

- Deep learning methods generally do NOT
  - incorporate intuitive rules such as negation
  - consider large-scale linguistic resources such as PPDB or WordNet

Methods

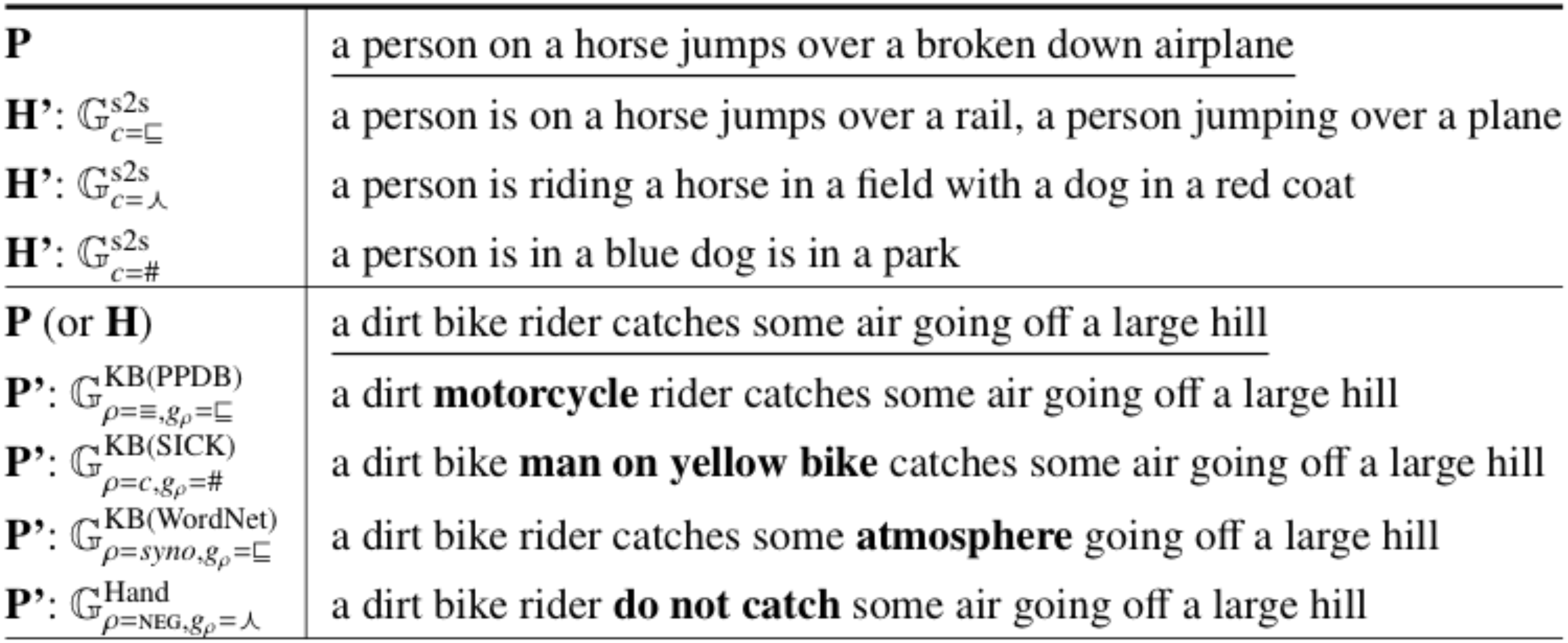
- How to do with Intuitive rules and linguistic resources
- Task-specific?
- Model-independent ?

Methods



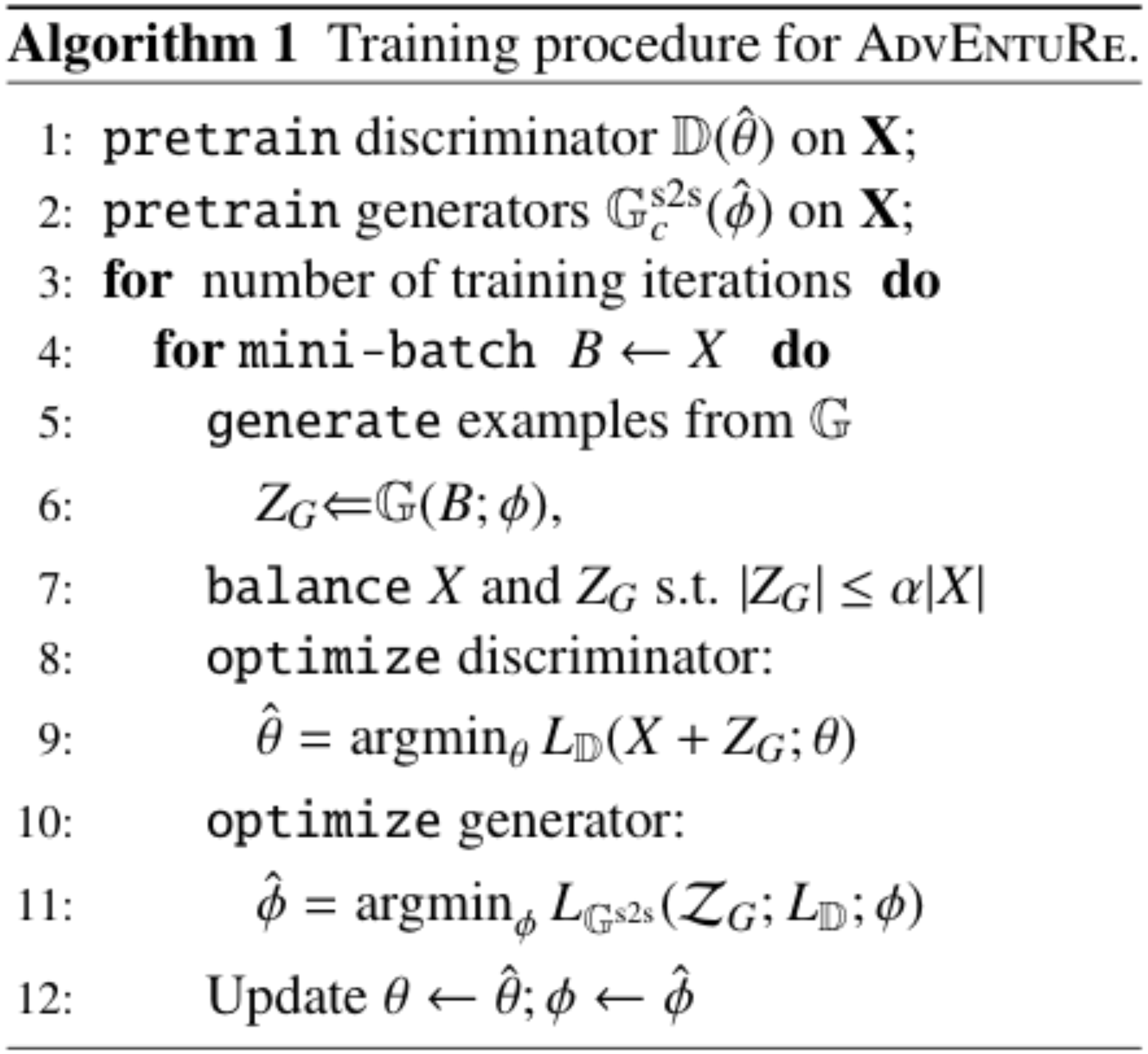
Source	ρ	f_ρ(s)	g_ρ
Knowledge Base, $\mathbb{G}^{\mathrm{KB}}$			
WordNet	hyper(x, y)	Replace x with y in s	$\sqsubseteq$
	anto(x, y)		$\curlywedge$
	syno(x, y)		$\sqsubseteq$
PPDB	$x \equiv y$		$\sqsubseteq$
SICK	$c(x, y)$		c
Hand-authored, $\mathbb{G}^{\mathrm{H}}$			
Domain knowledge	NEG	NEGATE(s)	$\curlywedge$
Neural Model, $\mathbb{G}^{\mathrm{s2s}}$			
Training data	(s2s, c)	$\mathbb{G}_c^{\mathrm{s2s}}(s)$	c

Methods



P	a person on a horse jumps over a broken down airplane
H': $\mathbb{G}^{\text{s2s}}_{c=\sqsubseteq}$	a person is on a horse jumps over a rail, a person jumping over a plane
H': $\mathbb{G}^{\text{s2s}}_{c=\text{人}}$	a person is riding a horse in a field with a dog in a red coat
H': $\mathbb{G}^{\text{s2s}}_{c=\#}$	a person is in a blue dog is in a park
P (or **H**)	a dirt bike rider catches some air going off a large hill
P': $\mathbb{G}^{\text{KB(PPDB)}}_{\rho=\equiv,g_\rho=\sqsubseteq}$	a dirt **motorcycle** rider catches some air going off a large hill
P': $\mathbb{G}^{\text{KB(SICK)}}_{\rho=c,g_\rho=\#}$	a dirt bike **man on yellow bike** catches some air going off a large hill
P': $\mathbb{G}^{\text{KB(WordNet)}}_{\rho=syno,g_\rho=\sqsubseteq}$	a dirt bike rider catches some **atmosphere** going off a large hill
P': $\mathbb{G}^{\text{Hand}}_{\rho=\text{NEG},g_\rho=\text{人}}$	a dirt bike rider **do not catch** some air going off a large hill

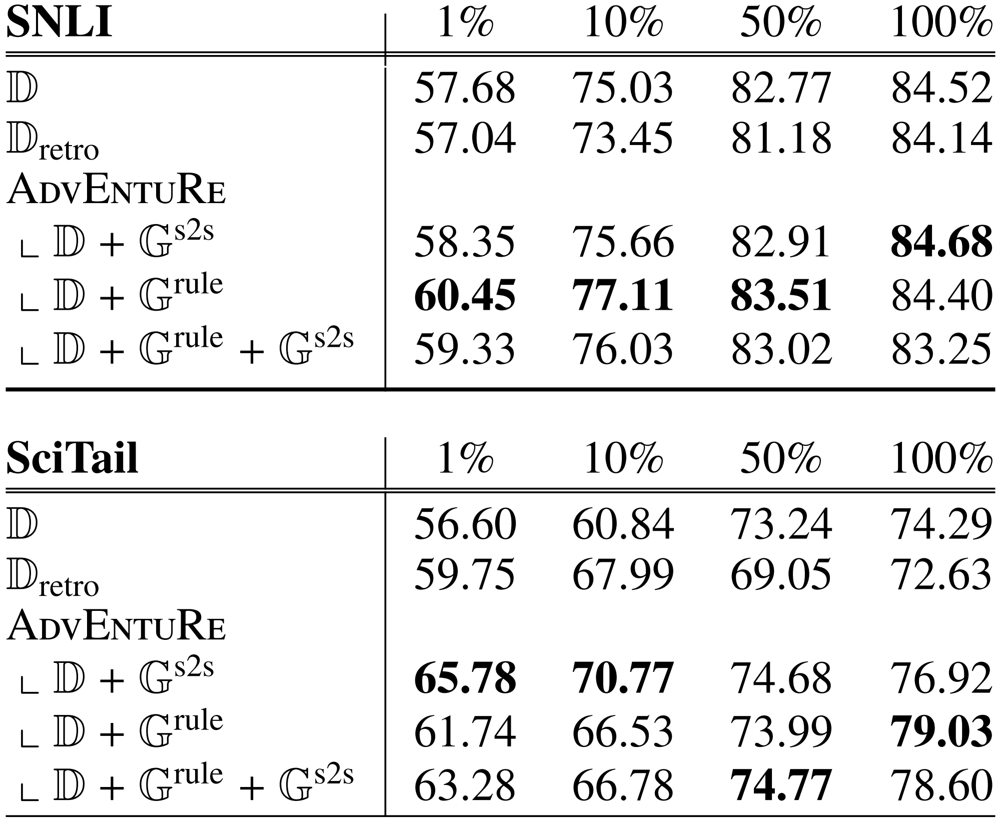
Methods



Algorithm 1 Training procedure for AdvEntuRe.

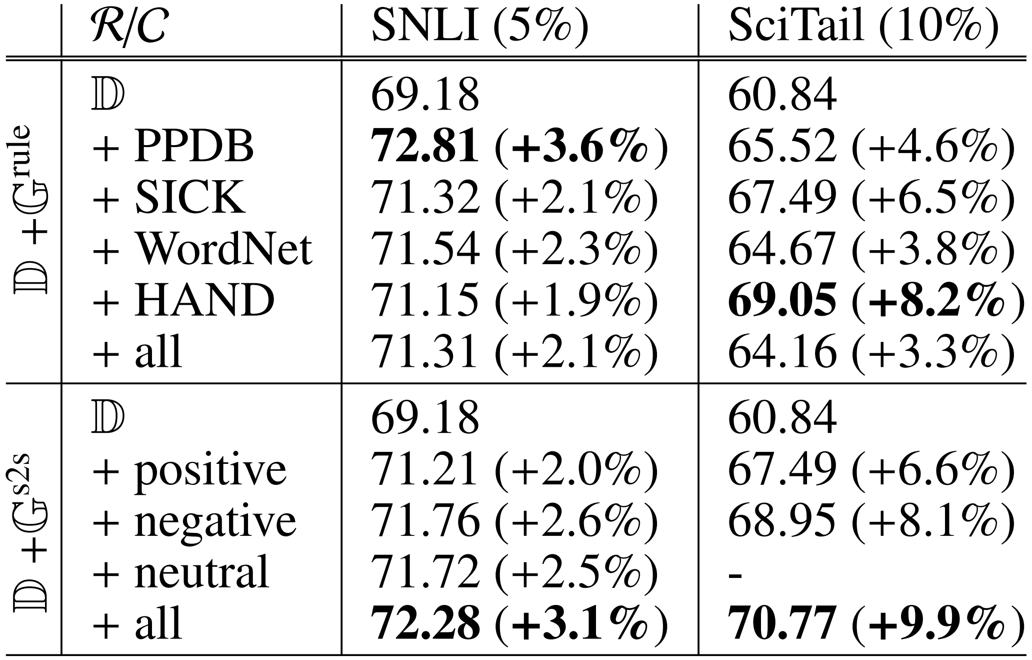
```
pretrain discriminator $\mathbb{D}(\hat{\theta})$ on $\mathbf{X}$;
pretrain generators $\mathbb{G}_c^{\text{s2s}}(\hat{\phi})$ on $\mathbf{X}$;
for number of training iterations do
    for mini-batch $B \leftarrow X$ do
        generate examples from $\mathbb{G}$
            $Z_G \Leftarrow \mathbb{G}(B; \phi)$,
        balance $X$ and $Z_G$ s.t. $|Z_G| \leq \alpha|X|$
        optimize discriminator:
            $\hat{\theta} = \text{argmin}_\theta\, L_{\mathbb{D}}(X + Z_G; \theta)$
        optimize generator:
            $\hat{\phi} = \text{argmin}_\phi\, L_{\mathbb{G}^{\text{s2s}}}(\mathcal{Z}_G; L_{\mathbb{D}}; \phi)$
        Update $\theta \leftarrow \hat{\theta}; \phi \leftarrow \hat{\phi}$
```

Experiments



SNLI	1%	10%	50%	100%
$\mathbb{D}$	57.68	75.03	82.77	84.52
$\mathbb{D}_{\text{retro}}$	57.04	73.45	81.18	84.14
AdvEntuRe				
⌞ $\mathbb{D}$ + $\mathbb{G}^{\text{s2s}}$	58.35	75.66	82.91	**84.68**
⌞ $\mathbb{D}$ + $\mathbb{G}^{\text{rule}}$	**60.45**	**77.11**	**83.51**	84.40
⌞ $\mathbb{D}$ + $\mathbb{G}^{\text{rule}}$ + $\mathbb{G}^{\text{s2s}}$	59.33	76.03	83.02	83.25

SciTail	1%	10%	50%	100%
$\mathbb{D}$	56.60	60.84	73.24	74.29
$\mathbb{D}_{\text{retro}}$	59.75	67.99	69.05	72.63
AdvEntuRe				
⌞ $\mathbb{D}$ + $\mathbb{G}^{\text{s2s}}$	**65.78**	**70.77**	74.68	76.92
⌞ $\mathbb{D}$ + $\mathbb{G}^{\text{rule}}$	61.74	66.53	73.99	**79.03**
⌞ $\mathbb{D}$ + $\mathbb{G}^{\text{rule}}$ + $\mathbb{G}^{\text{s2s}}$	63.28	66.78	**74.77**	78.60



	$\mathcal{R}/C$	SNLI (5%)	SciTail (10%)
$\mathbb{D}$ +$\mathbb{G}^{\text{rule}}$	$\mathbb{D}$	69.18	60.84
	+ PPDB	**72.81** (**+3.6%**)	65.52 (+4.6%)
	+ SICK	71.32 (+2.1%)	67.49 (+6.5%)
	+ WordNet	71.54 (+2.3%)	64.67 (+3.8%)
	+ HAND	71.15 (+1.9%)	**69.05** (**+8.2%**)
	+ all	71.31 (+2.1%)	64.16 (+3.3%)
$\mathbb{D}$ +$\mathbb{G}^{\text{s2s}}$	$\mathbb{D}$	69.18	60.84
	+ positive	71.21 (+2.0%)	67.49 (+6.6%)
	+ negative	71.76 (+2.6%)	68.95 (+8.1%)
	+ neutral	71.72 (+2.5%)	-
	+ all	**72.28** (**+3.1%**)	**70.77** (**+9.9%**)

Learn a Lesson

- Easiest way to do a **good** but not exciting work
  - Find right problem
  - do trivial but **promising** ideas
  - make **elaborate** experiment analysis.