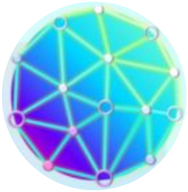
Shoring up the Foundations: Fusing Weak Supervision and Model Embeddings

Mayee F. Chen*, Daniel Y. Fu*, Dyah Adila, Michael Zhang, Fred Sala, Kayvon Fatahalian, Christopher Ré

August 2, 2022, UAI

Motivation

Foundation Models (FMs)[1]



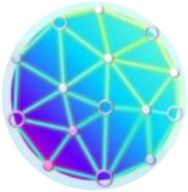
E.g. GPT3, CLIP, PaLM

\+ Perform extremely well on variety of downstream tasks

[1] Bommasani et. al. On the Opportunities and Risks of Foundation Models, 2021.

Motivation

Foundation Models (FMs)[1]



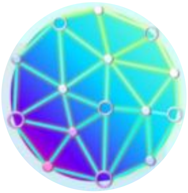
E.g. GPT3, CLIP, PaLM

+ Perform extremely well on variety of downstream tasks
- Fixed interfaces, may require hand-labeled data to adapt to task

[1] Bommasani et. al. On the Opportunities and Risks of Foundation Models, 2021.

Motivation

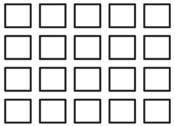
Foundation Models (FMs)[1]



E.g. GPT3, CLIP, PaLM

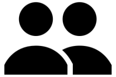
+ Perform extremely well on variety of downstream tasks
- Fixed interfaces, may require hand-labeled data to adapt to task

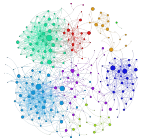
Weak Supervision (WS)



Unlabeled data

+





Crowdworkers, heuristics, external KBs

=

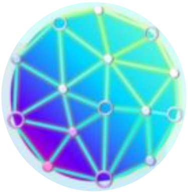
WS data labeling pipelines in Google, Youtube; startup Snorkel AI[2]

+ Produce labeled data using unlabeled data and noisier, weaker sources

[1] Bommasani et. al. On the Opportunities and Risks of Foundation Models, 2021.
[2] snorkel.ai

Motivation

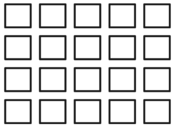
Foundation Models (FMs)[1]



E.g. GPT3, CLIP, PaLM

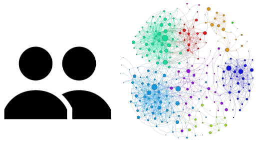
+ Perform extremely well on variety of downstream tasks
- Fixed interfaces, may require hand-labeled data to adapt to task

Weak Supervision (WS)



Unlabeled data

\+



Crowdworkers, heuristics, external KBs

= *WS data labeling pipelines in Google, Youtube; startup Snorkel AI*[2]

+ Produce labeled data using unlabeled data and noisier, weaker sources
- Source quality can vary significantly across data

[1] Bommasani et. al. On the Opportunities and Risks of Foundation Models, 2021.
[2] snorkel.ai

Q: How can we combine Foundation Models and Weak Supervision in settings where we lack hand-labeled data?

Q: How can we combine Foundation Models and Weak Supervision in settings where we lack hand-labeled data?

Our work: *exploit smoothness of FM embeddings to address particular algorithmic challenges in Weak Supervision*

Outline

1. Problem Setup
   a. Simple baselines combining WS and FMs
2. Technical Overview of Weak Supervision
   a. Background
   b. 2 Challenges → potential FM interface opportunities
3. Method (Liger): using FMs to solve WS challenges
4. Theory: Embedding Smoothness
5. Results

Problem Setup

Formal Problem Setup

Input:

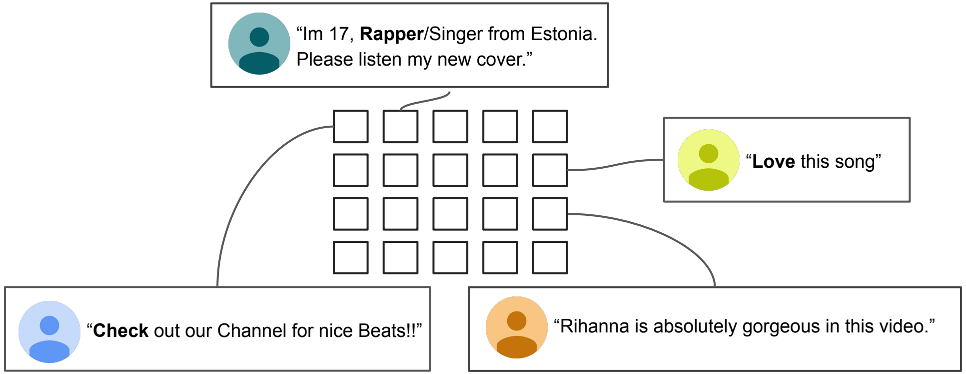
- Unlabeled dataset $\mathcal{D} = \{x_i\}_{i=1}^{n}$ with unknown label $y \in \{-1, 1\}$
- Weak sources' **labeling functions** (LFs) $\lambda_1, \ldots, \lambda_m : \mathcal{X} \to \{-1, 0, 1\}$
    - Sources can *abstain* and output 0

Formal Problem Setup

Input:

- Unlabeled dataset $\mathcal{D} = \{x_i\}_{i=1}^{n}$ with unknown label $y \in \{-1, 1\}$
- Weak sources' **labeling functions** (LFs) $\lambda_1, \ldots, \lambda_m : \mathcal{X} \rightarrow \{-1, 0, 1\}$
    - Sources can *abstain* and output 0

Example: YouTube spam comment dataset



Spam LFs

λ_1
```
def L_1:
   NOT SPAM if "love"
```

λ_2
```
def L_2:
   SPAM if "rapper"
```

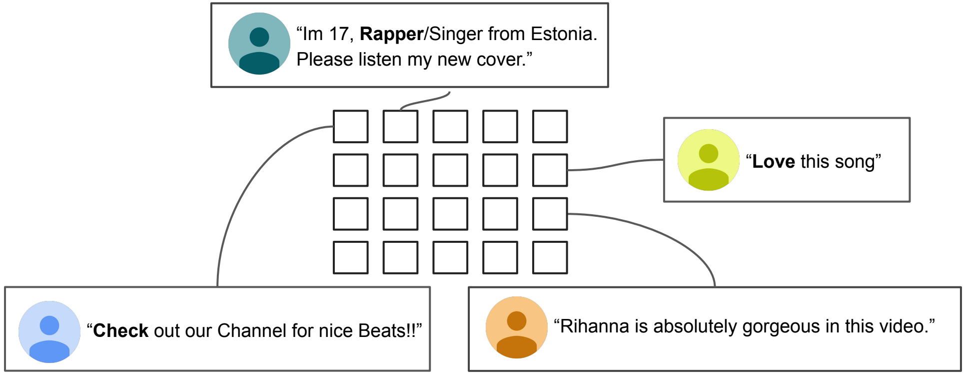
λ_3
```
def L_3:
   SPAM if "check out"
```

Formal Problem Setup

Input:

- Unlabeled dataset $\mathcal{D} = \{x_i\}_{i=1}^{n}$ with unknown label $y \in \{-1, 1\}$
- Weak sources' **labeling functions** (LFs) $\lambda_1, \ldots, \lambda_m : \mathcal{X} \to \{-1, 0, 1\}$
  - Sources can *abstain* and output 0
- FM embedding mapping $f : \mathcal{X} \to \mathbb{R}^d$
  - Model accessible via embeddings (e.g. OpenAI API), cannot access full model

Example: YouTube spam comment dataset



Spam LFs

λ_1
```
def L_1:
   NOT SPAM if “love”
```

λ_2
```
def L_2:
   SPAM if “rapper”
```

λ_3
```
def L_3:
   SPAM if “check out”
```

Formal Problem Setup

Input:

- Unlabeled dataset $\mathcal{D} = \{x_i\}_{i=1}^{n}$ with unknown label $y \in \{-1, 1\}$
- Weak sources' **labeling functions** (LFs) $\lambda_1, \ldots, \lambda_m : \mathcal{X} \to \{-1, 0, 1\}$
  - Sources can *abstain* and output 0
- FM embedding mapping $f : \mathcal{X} \to \mathbb{R}^d$
  - Model accessible via embeddings (e.g. OpenAI API), cannot access full model

Desired output: $\Pr_f(y = 1 | \lambda_1, \ldots, \lambda_m, x)$

- Given a datapoint, a list of votes on its label, and its embedding, what is its true label?

Some Simple Baselines

Use weak supervision and FM embeddings *sequentially*?

Unlabeled dataset

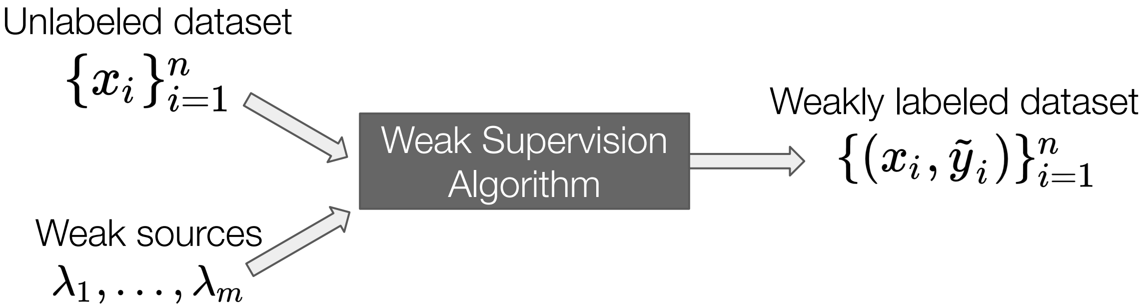
$$\{x_i\}_{i=1}^{n}$$

Weak sources

$$\lambda_1, \ldots, \lambda_m$$

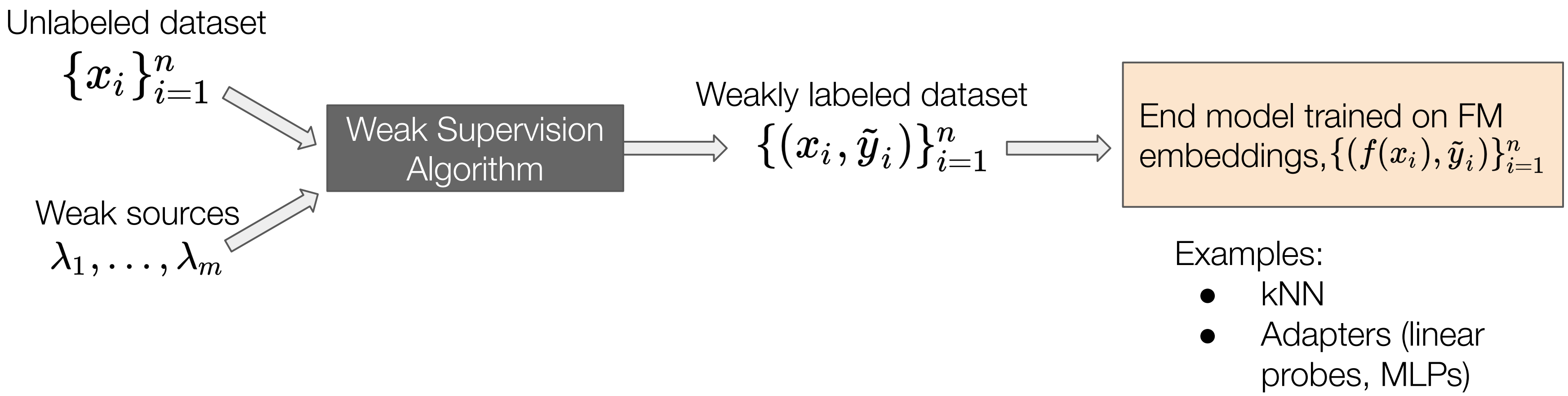
Some Simple Baselines

Use weak supervision and FM embeddings *sequentially?*



Some Simple Baselines

Use weak supervision and FM embeddings *sequentially?*



Some Simple Baselines

Use weak supervision and FM embeddings *sequentially?*

Unlabeled dataset $\{x_i\}_{i=1}^n$

Weak sources $\lambda_1, \ldots, \lambda_m$

$\rightarrow$ **Weak Supervision Algorithm** $\rightarrow$ Weakly labeled dataset $\{(x_i, \tilde{y}_i)\}_{i=1}^n$ $\rightarrow$ End model trained on FM embeddings, $\{(f(x_i), \tilde{y}_i)\}_{i=1}^n$

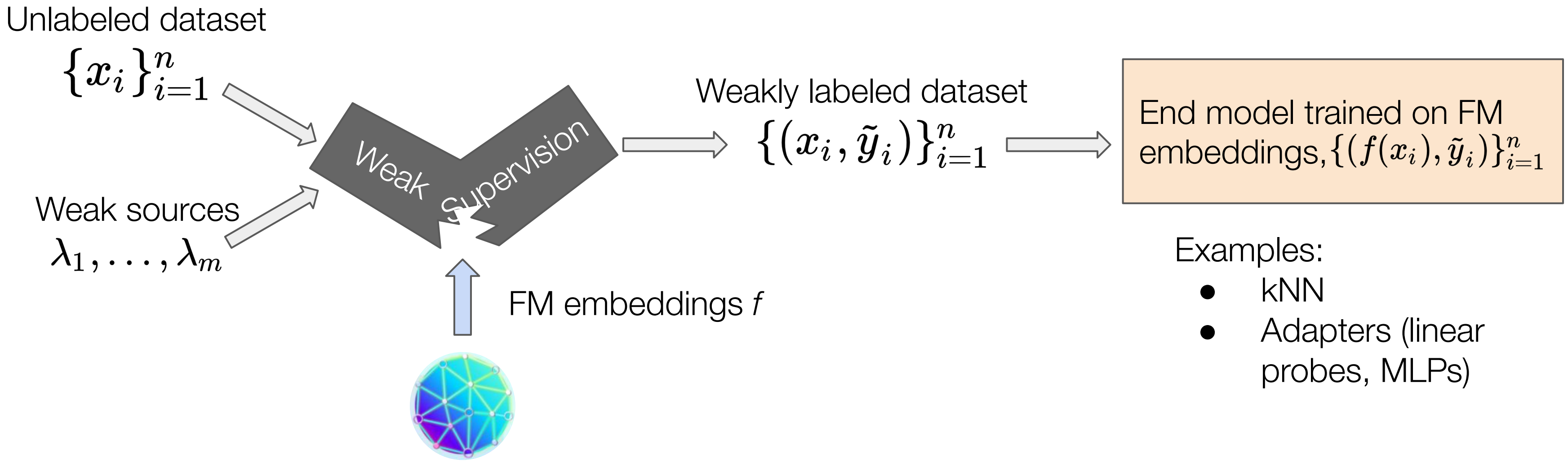
Examples:

- kNN
- Adapters (linear probes, MLPs)

Can we do better than sequential application?

Some Simple Baselines

Use weak supervision and FM embeddings *sequentially?*



Examples:

- kNN
- Adapters (linear probes, MLPs)

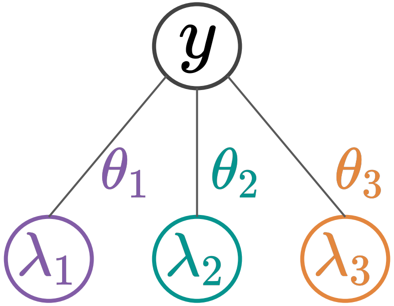
Can we do better than sequential application?

Deeper Dive into Weak Supervision

Standard WS Algorithm

1. Learn relationship between $\boldsymbol{y}$ and $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_m$ via *graphical model*

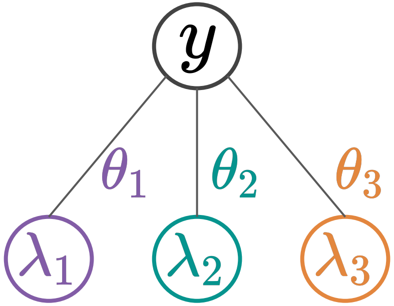
$$\Pr_\theta(y, \lambda_1, \ldots, \lambda_m) \Longleftrightarrow$$



Standard WS Algorithm

1. Learn relationship between y and $\lambda_1, \ldots, \lambda_m$ via *graphical model*

$$\Pr_\theta(y, \lambda_1, \ldots, \lambda_m) \Longleftrightarrow$$



- Learn accuracy parameters $\theta_i = \mathbb{E}[\lambda_i y]$
- Under the hood: algorithms for *latent variable estimation*[1,2] — compute how often LFs agree on their votes (covariance matrix) — or via maximum likelihood estimation[3]

[1] Fu et. al. Fast and three-rious: Speeding up weak supervision with triplet methods. ICML 2020.
[2] Ratner et. al. Training Complex Models with Multi-Task Weak Supervision. AAAI 2019.
[3] Ratner et. al. Data Programming: Creating Large Training Sets, Quickly. NeurIPS 2016.

Standard WS Algorithm

2. Inference

- Given *x*, output estimate of $\mathbf{Pr}_\theta(y = 1|\lambda_1, \ldots, \lambda_m)$
- Under the hood: Bayes rule, weighing each λ_i by a function of θ_i

Two Challenges

Coarse-grained accuracies

$\mathbf{Pr}_\theta(y = 1|\lambda_1, \ldots, \lambda_m)$ is independent of *x*

Two Challenges

Coarse-grained accuracies

$\Pr_\theta(y = 1|\lambda_1, \ldots, \lambda_m)$ is independent of *x*

x_1: spam

"Im 17, **Rapper**/Singer from Estonia. Please listen my new cover."

x_2: not spam

love the way you lie featuring rhianna, hes an awesome **rapper**!!! shes an awesome singer!!!

```
def L_2:
    SPAM if "rapper"
```

$$\lambda_2(x_1) = \lambda_2(x_2)$$

Model output is the same for both points, ignoring other contextual information in the comments

Two Challenges

Coarse-grained accuracies

$\mathrm{Pr}_\theta(y = 1|\lambda_1, \ldots, \lambda_m)$ is independent of *x*

x_1: spam

“Im 17, **Rapper**/Singer from Estonia. Please listen my new cover.”

x_2: not spam

love the way you lie featuring rhianna, hes an awesome **rapper**!!! shes an awesome singer!!!

λ_2 is accurate here

λ_2 is inaccurate here

```
def L_2:
   SPAM if “rapper”
```

$\lambda_2(x_1) = \lambda_2(x_2)$

But we only have one accuracy parameter θ_2 for λ_2!

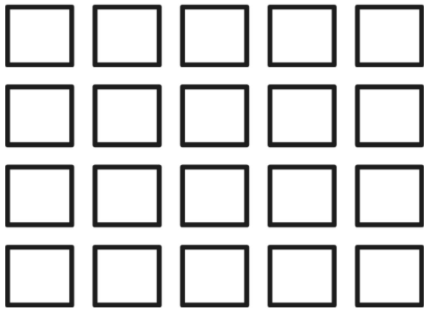
Model output is the same for both points, ignoring other contextual information in the comments

Two Challenges

Low coverage

When $\lambda_i = 0$ on *x*, the algorithm discards that LF because it is uninformative.

- Outputs $\mathrm{Pr}_\theta(y = 1 | \lambda_i, \ldots, \lambda_{i-1}, \lambda_{i+1}, \lambda_m)$ on *x*

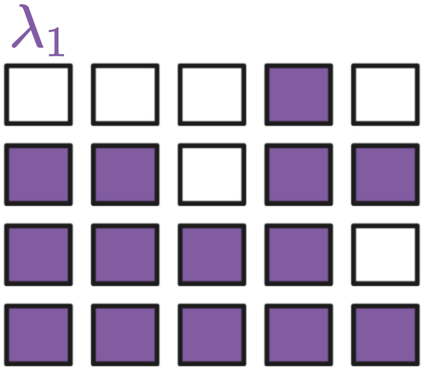


Two Challenges

Low coverage

When $\lambda_i = 0$ on *x*, the algorithm discards that LF because it is uninformative.

- Outputs $\Pr_\theta(y = 1 | \lambda_i, \ldots, \lambda_{i-1}, \lambda_{i+1}, \lambda_m)$ on *x*

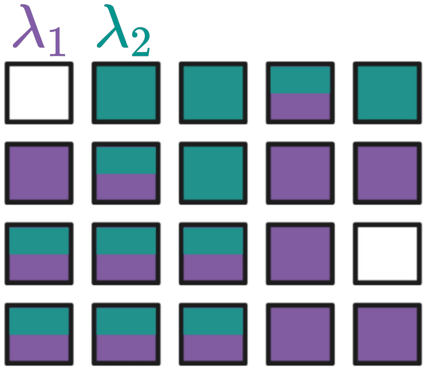


Two Challenges

Low coverage

When $\lambda_i = 0$ on *x*, the algorithm discards that LF because it is uninformative.

- Outputs $\Pr_\theta(y = 1 | \lambda_i, \ldots, \lambda_{i-1}, \lambda_{i+1}, \lambda_m)$ on *x*

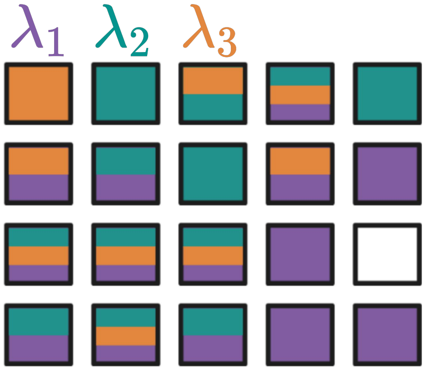


Two Challenges

Low coverage

When $\lambda_i = 0$ on *x*, the algorithm discards that LF because it is uninformative.

- Outputs $\Pr_\theta(y = 1 | \lambda_i, \ldots, \lambda_{i-1}, \lambda_{i+1}, \lambda_m)$ on *x*



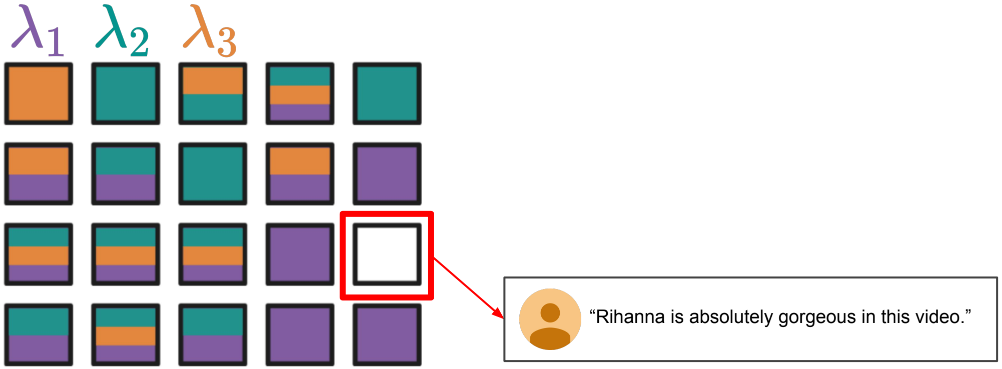
Two Challenges

Low coverage

When $\lambda_i = 0$ on *x*, the algorithm discards that LF because it is uninformative.

- Outputs $\Pr_\theta(y = 1 | \lambda_i, \ldots, \lambda_{i-1}, \lambda_{i+1}, \lambda_m)$ on *x*

low *coverage* of LFs
= some points lack signal, bad WS output



No "check out", "love" or "rapper" in this comment

Method

Liger 🦁🐯

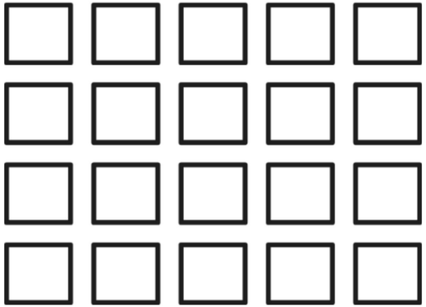
We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

1. For coarse-grained accuracies, **partition** the FM embedding space and estimate a set of parameters per part → finer grained accuracies, better estimate of $\mathbf{Pr}_\theta(y = 1|\lambda_1, \ldots, \lambda_m, x)$

Dataset $\{x_i\}_{i=1}^n$

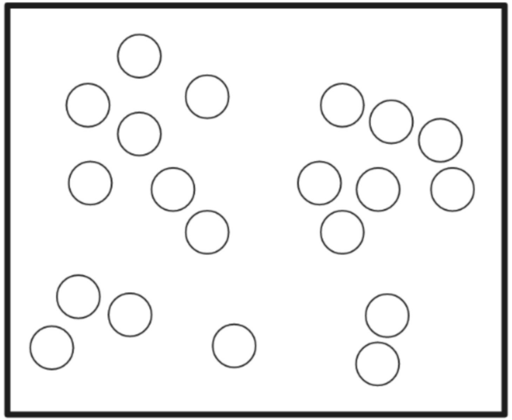


Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

1. For coarse-grained accuracies, **partition** the FM embedding space and estimate a set of parameters per part → finer grained accuracies, better estimate of $\mathbf{Pr}_\theta(y = 1|\lambda_1, \ldots, \lambda_m, x)$

Embedded dataset $\{f(x_i)\}_{i=1}^n$

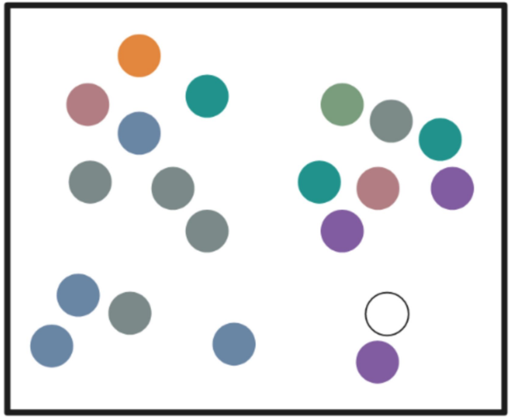


Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

1. For coarse-grained accuracies, **partition** the FM embedding space and estimate a set of parameters per part → finer grained accuracies, better estimate of $\mathbf{Pr}_\theta(y = 1|\lambda_1, \ldots, \lambda_m, x)$

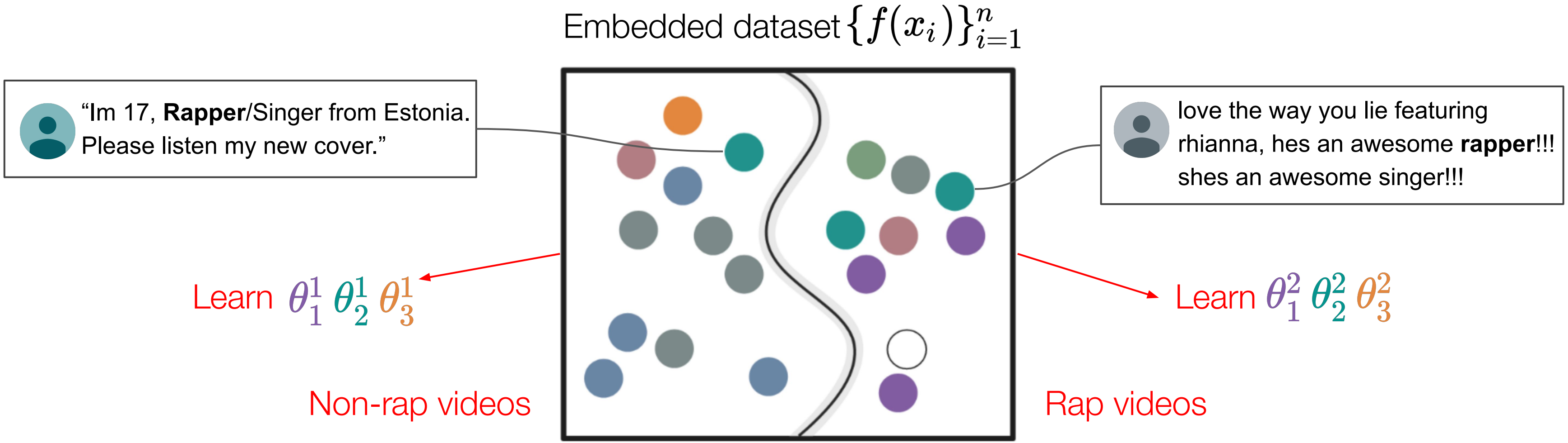
Embedded dataset $\{f(x_i)\}_{i=1}^n$



Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

1. For coarse-grained accuracies, **partition** the FM embedding space and estimate a set of parameters per part → finer grained accuracies, better estimate of $\mathbf{Pr}_\theta(y = 1|\lambda_1, \ldots, \lambda_m, x)$

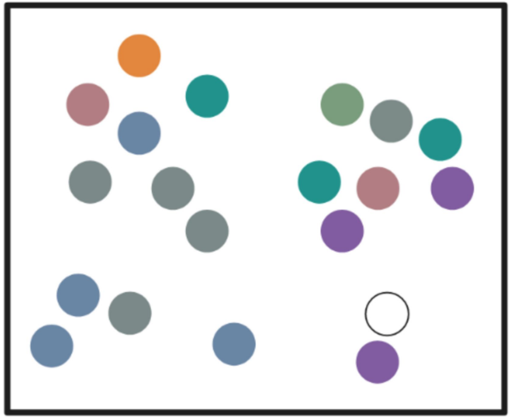


Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

2. For low coverage, **extend** votes of LFs to points that are close by in FM embedding space to construct $\bar{\lambda}_1, \ldots \bar{\lambda}_m$ → fewer abstains

Embedded dataset $\{f(x_i)\}_{i=1}^n$

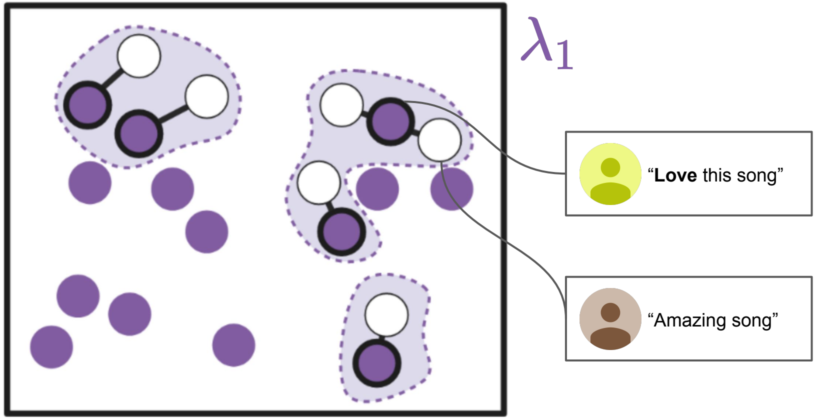


Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

2. For low coverage, **extend** votes of LFs to points that are close by in FM embedding space to construct $\bar{\lambda}_1, \ldots \bar{\lambda}_m \rightarrow$ fewer abstains

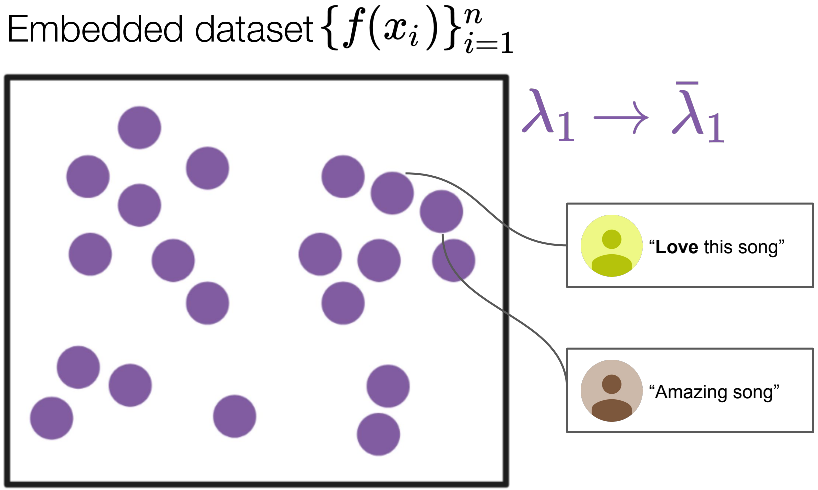
Embedded dataset $\{f(x_i)\}_{i=1}^n$



Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

2. For low coverage, **extend** votes of LFs to points that are close by in FM embedding space to construct $\bar{\lambda}_1, \ldots \bar{\lambda}_m$ → fewer abstains

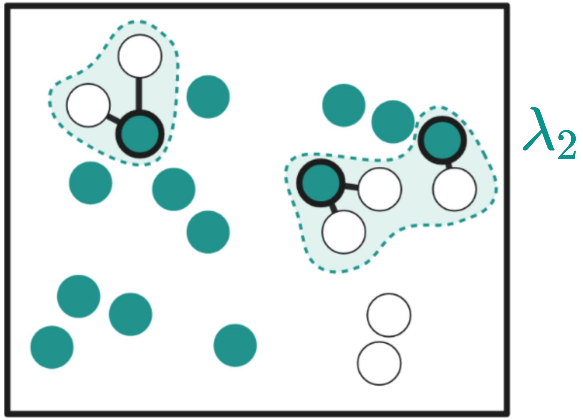


Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

2. For low coverage, **extend** votes of LFs to points that are close by in FM embedding space to construct $\bar{\lambda}_1, \ldots \bar{\lambda}_m$ → fewer abstains

Embedded dataset $\{f(x_i)\}_{i=1}^n$

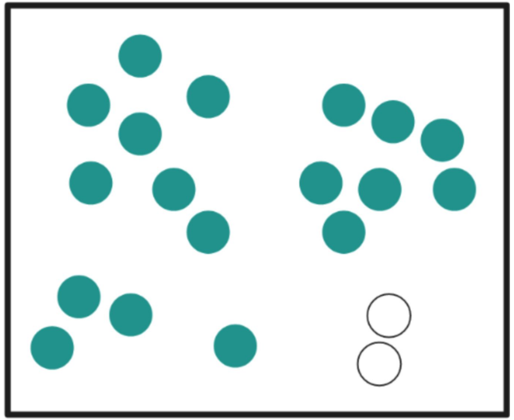


Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

2. For low coverage, **extend** votes of LFs to points that are close by in FM embedding space to construct $\bar{\lambda}_1, \ldots \bar{\lambda}_m$ → fewer abstains

Embedded dataset $\{f(x_i)\}_{i=1}^n$



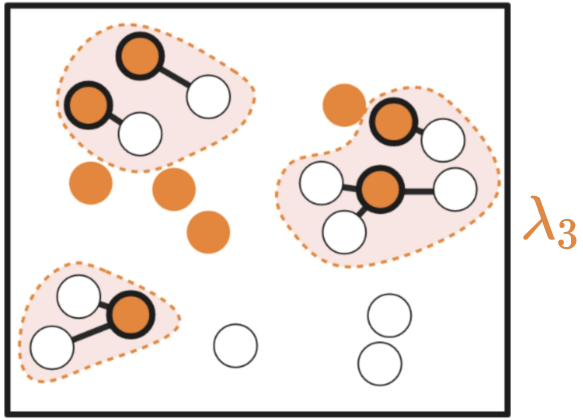
$\lambda_2 \rightarrow \bar{\lambda}_2$

Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

2. For low coverage, **extend** votes of LFs to points that are close by in FM embedding space to construct $\bar{\lambda}_1, \ldots \bar{\lambda}_m$ → fewer abstains

Embedded dataset $\{f(x_i)\}_{i=1}^n$

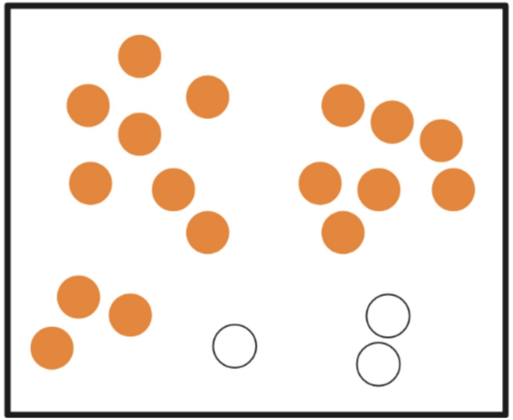


Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

2. For low coverage, **extend** votes of LFs to points that are close by in FM embedding space to construct $\bar{\lambda}_1, \ldots \bar{\lambda}_m$ $\rightarrow$ fewer abstains

Embedded dataset $\{f(x_i)\}_{i=1}^n$



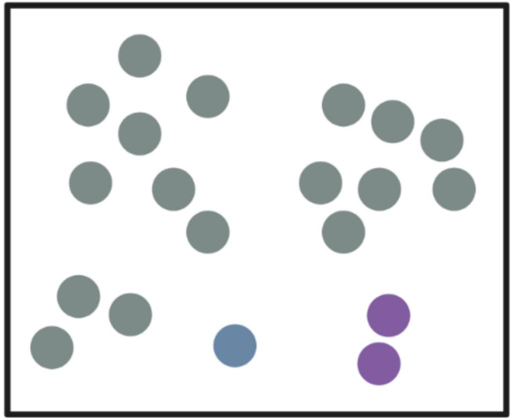
$\lambda_3 \rightarrow \bar{\lambda}_3$

Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

2. For low coverage, **extend** votes of LFs to points that are close by in FM embedding space to construct $\bar{\lambda}_1, \dots \bar{\lambda}_m \rightarrow$ fewer abstains

Embedded dataset $\{f(x_i)\}_{i=1}^{n}$



$\lambda_1 \rightarrow \bar{\lambda}_1$

$\lambda_2 \rightarrow \bar{\lambda}_2$

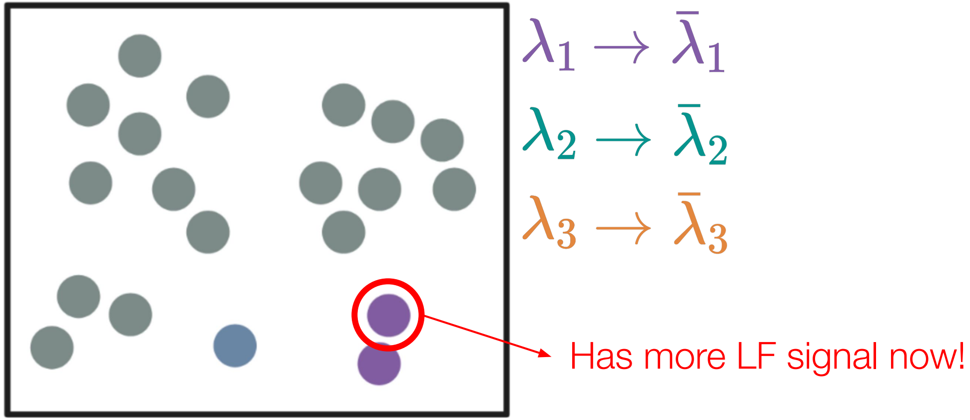
$\lambda_3 \rightarrow \bar{\lambda}_3$

Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

2. For low coverage, **extend** votes of LFs to points that are close by in FM embedding space to construct $\bar{\lambda}_1, \ldots \bar{\lambda}_m$ $\rightarrow$ fewer abstains

Embedded dataset $\{f(x_i)\}_{i=1}^n$



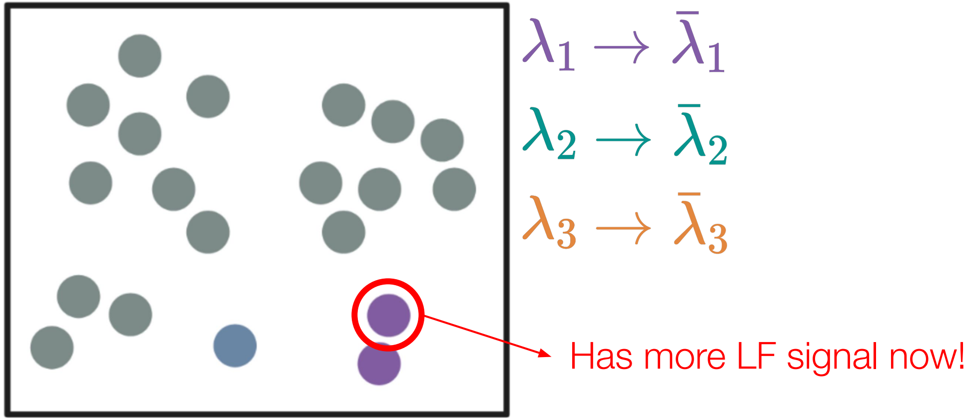
Liger 🦁🐯

We address the 2 challenges by viewing them as *2 opportunities for interfacing with foundation models*:

2. For low coverage, **extend** votes of LFs to points that are close by in FM embedding space to construct $\bar{\lambda}_1, \ldots \bar{\lambda}_m$ → fewer abstains

Altogether: estimate $\Pr(y = 1 | \bar{\lambda}_1, \ldots, \bar{\lambda}_m, x)$ using partitioned dataset based off of *f*

Embedded dataset $\{f(x_i)\}_{i=1}^n$



Why does this work?

Theory: why does this work?

Liger works because of the local *smoothness* of labels in the FM embedding space.

Theory: why does this work?

Liger works because of the local *smoothness* of labels in the FM embedding space.

Smoothness (informal): how unlikely the label changes as you move further away from a point

- E.g., $\Pr(y = 1|x) - \Pr(y = 1|x') \leq K||f(x) - f(x')||$

Theory: why does this work?

Liger works because of the local *smoothness* of labels in the FM embedding space.

Smoothness (informal): how unlikely the label changes as you move further away from a point

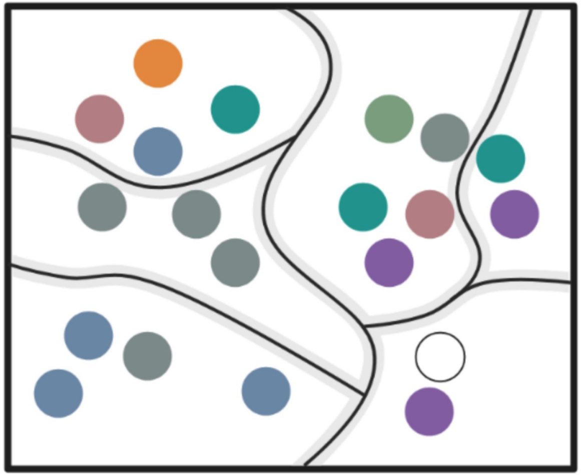
- E.g., $\Pr(y = 1|x) - \Pr(y = 1|x') \leq K\|f(x) - f(x')\|$

Smoothness allows for:

- Good approximation of Pr(|x) when partitioning
- Extended labeling functions to be accurate nearby

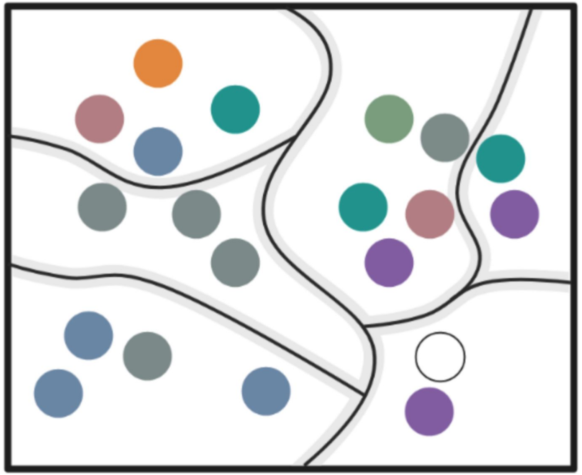
Theory: tradeoffs

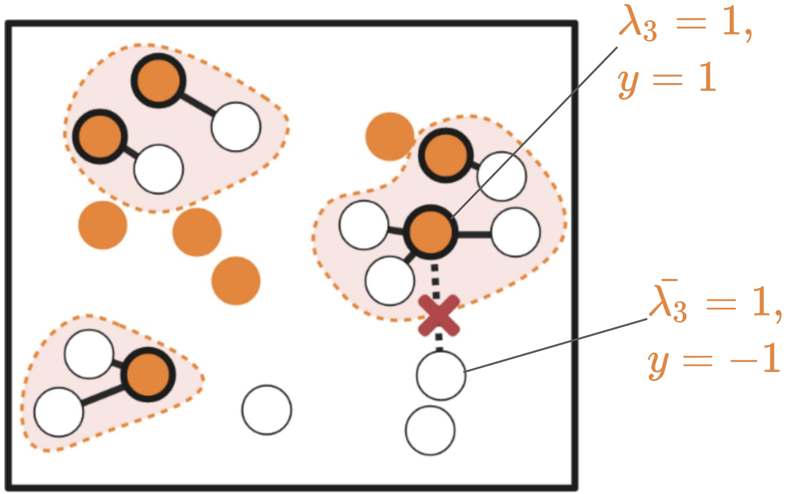
- ⚠️ Partitioning into too many sets = good local estimates, high variance
- ⚠️ Extending too far in embedding space = LF votes become incorrect because true label changes value
- Need to control these depending on how smooth FM is!



Theory: tradeoffs

- ⚠️ Partitioning into too many sets = good local estimates, high variance
- ⚠️ Extending too far in embedding space = LF votes become incorrect because true label changes value
- Need to control these depending on how smooth FM is!





Theoretical results (informal)

Theorem 1: generalization error of Liger has a bias-variance decomposition dependent on size/number of partitions and a smoothness constant.

- Tradeoff: more partitioning = lower bias, higher variance

Theoretical results (informal)

Theorem 1: generalization error of Liger has a bias-variance decomposition dependent on size/number of partitions and a smoothness constant.

- Tradeoff: more partitioning = lower bias, higher variance

Theorem 2: extending labeling functions further reduces generalization error if $\bar{\lambda}$'s accuracy is better than random. This can be achieved depending on λ's accuracy, FM smoothness, and how much we extend.

- Tradeoff: more extending = more coverage, worse accuracy

Empirical Results

Validation

1. No hand-labeled data: compare against a) WS (no FMs), b) sequential WS+FM baselines
2. [In paper] some labeled data: can we combine LIGER with labeled data?
3. Is embedding smoothness correlated with performance?

Empirical Results (no hand-labeled data)

Weak Supervision Datasets + GPT-3 embeddings (NLP), CLIP embeddings (video)

	Task	WS-kNN	WS-Adapter	WS-LM	LIGER	ΔCoverage
		Weak Sources Only				
NLP	**Spam**	72.8	92.3	83.6	**95.0**	+45.5
	Weather	62.0	86.0	78.0	**98.0**	+90.2
	Spouse	16.9	17.1	47.0	**52.2**	+12.1
Video	**Basketball**	33.3	48.9	27.9	**69.6**	+8.3
	Commercial	84.7	92.8	88.4	**93.5**	+18.8
	Tennis	83.0	**83.8**	82.0	83.3	+32.5

Without additional hand-labeled data, Liger improves LF coverage and outperforms standard WS and sequential baselines

Empirical Results (no hand-labeled data)

Weak Supervision Datasets + GPT-3 embeddings (NLP), CLIP embeddings (video)

Task	Standard WS	WS + kNN	WS + Adapter	**Liger**	ΔCoverage
Spam	83.6	72.8	<u>92.3</u>	**95.0**	+45.5
Weather	78.0	62.0	<u>86.0</u>	**98.0**	+90.2
Spouse	<u>47.0</u>	16.9	17.1	**52.2**	+12.1
Basketball	27.9	33.3	<u>48.9</u>	**69.6**	+8.3
Commercial	88.4	84.7	<u>92.8</u>	**93.5**	+18.8
Tennis	82.0	83.0	**83.8**	<u>83.3</u>	+32.5

Without additional hand-labeled data or end model training, Liger improves LF coverage and outperforms standard WS and sequential baselines

Empirical Results (some labeled data)

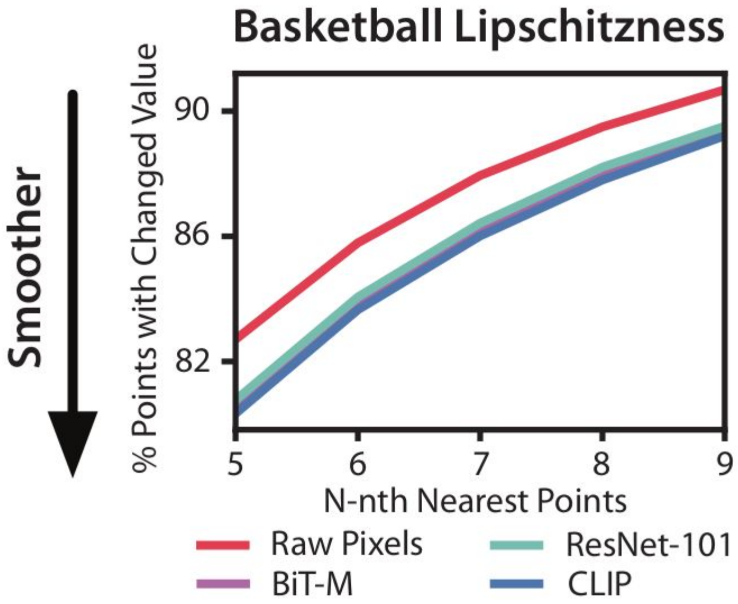
What if we have a little bit of labeled data available?

Liger-adapter: use Liger model outputs + labeled data as input to adapter

Task	kNN	Adapter	**Liger-Adapter**
Spam	91.2	94.4	**95.4**
Weather	92.0	90.0	**96.8**
Spouse	21.6	15.7	**49.6**
Basketball	64.4	79.3	**79.5**
Commercial	92.0	93.0	**93.2**
Tennis	73.2	83.1	**84.0**

Liger-adapter allows for our method to incorporate labeled data and outperforms baselines that do not utilize unlabeled data

A closer look at smoothness



Embedding	F1-score
Raw pixel	19.3
RN-101	31.1
BiT-M	42.5
CLIP	**69.6**

+ Explore smoothness of prompting methods for text

Matches theory that more smooth FM embeddings = better performance

Summary

- Liger applies foundation models to weak supervision settings, which lack hand-labeled data, with two simple steps that exploit FM *smoothness*
  - **Partition in FM embedding space:** Estimate finer-grained accuracy parameters
  - **Extend in FM embedding space:** Improve coverage of labeling functions
- Improves over standard WS and simple baselines based on FM smoothness

Summary

- Liger applies foundation models to weak supervision settings, which lack hand-labeled data, with two simple steps that exploit FM *smoothness*
    - **Partition in FM embedding space:** Estimate finer-grained accuracy parameters
    - **Extend in FM embedding space:** Improve coverage of labeling functions
- Improves over standard WS and simple baselines based on FM smoothness

Takeaway: Liger shows that the application of foundation model interfaces can be algorithm-aware and principled (e.g. via smoothness property)

- But there may be many more interesting ways to combine FMs and weak supervision principles!

Thank you!

Contact: mfchen@stanford.edu, danfu@cs.stanford.edu

Arxiv: https://arxiv.org/abs/2203.13270

Code: https://github.com/HazyResearch/liger

Mayee F. Chen*, Daniel Y. Fu*, Dyah Adila, Michael Zhang, Fred Sala, Kayvon Fatahalian, Christopher Ré. *Shoring Up the Foundations: Fusing Model Embeddings and Weak Supervision.* UAI 2022.






Theoretical Results

Thm 1 (informal). Suppose we partition data into *s* sets, and *d* is the average diameter of a set in embedding space. *K* is a smoothness constant (lower *K* = more smooth). Then, Liger's error (no extensions) is:

$$\underbrace{Error \leq K \cdot d}_{\text{Bias}} + \underbrace{\mathcal{O}\left(\frac{ms}{n}\right)}_{\text{Variance}} + \underbrace{H(y|\lambda_1, \ldots \lambda_m, x)}_{\substack{\text{Irreducible Error} \\ \text{(conditional entropy)}}}$$

- Bias-variance tradeoff in size/number of partitions, depending on smoothness
- Irreducible error: amount of randomness in *y* after observing *x* and LF votes

Theoretical Results

Next, what does extending and using $\bar{\lambda}$ do?

- Decreases variance (more coverage = more points to estimate on)
- Irreducible error: $H(y|\lambda, x) \to H(y|\bar{\lambda}, x)$ unclear!

Theoretical Results

Next, what does extending and using $\bar{\lambda}$ do?

- Decreases variance (more coverage = more points to estimate on)
- Irreducible error: $H(y|\lambda, x) \to H(y|\bar{\lambda}, x)$ unclear!

Thm 2 (informal): $H(y|\bar{\lambda}, x) < H(y|\lambda, x)$ as long as $\bar{\lambda}$ has better-than-random accuracy. This can be achieved depending on λ's accuracy, FM smoothness, and how much we extend.

Theoretical Results

Next, what does extending and using $\bar{\lambda}$ do?

- Decreases variance (more coverage = more points to estimate on)
- Irreducible error: $H(y|\lambda, x) \to H(y|\bar{\lambda}, x)$ unclear!

Thm 2 (informal): $H(y|\bar{\lambda}, x) < H(y|\lambda, x)$ as long as $\bar{\lambda}$ has better-than-random accuracy. This can be achieved depending on λ's accuracy, FM smoothness, and how much we extend.

- Tradeoff: extending too much = worse accuracy, higher coverage

Empirical Results (some labeled data)

What if we have a little bit of labeled data available?

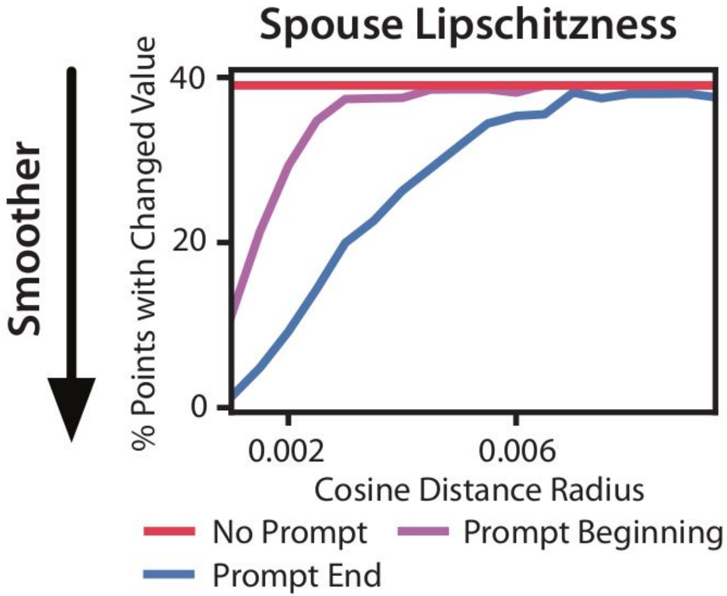
Liger-adapter: use Liger model outputs + labeled data as input to adapter

		Dev Labels Available		
	Task	**kNN**	**Adapter**	**LIGER-Adapter**
NLP	**Spam**	91.2	94.4	**95.4**
	Weather	92.0	90.0	**96.8**
	Spouse	21.6	15.7	**49.6**
Video	**Basketball**	64.4	79.3	**79.5**
	Commercial	92.0	93.0	**93.2**
	Tennis	73.2	83.1	**84.0**

Liger-adapter allows for our method to incorporate labeled data and outperforms baselines that do not utilize unlabeled data

A closer look at smoothness

What’s the best way to embed sentences?



Prompting	F1-score
No Prompt	48.5
Prompt Beginning	50.2
Prompt End	**52.2**

Matches theory that more smooth FM embeddings = better performance