Text as Data: Using LLMs for Annotation

Guest Course – January 2026

Germain Gauthier, Philine Widmer[1]

[1]Bocconi University, Paris School of Economics

USI Lugano

Lecture Outline

Part I: Validation Framework

1. Why validate LLM annotations?
2. LLM annotation setup & results
3. Human validation
4. Validation with labeled data
5. External proxy validation

Part II: Running LLMs

6. API vs. local deployment
7. Prompting strategies
8. Fine-tuning considerations
9. Privacy & guardrails
10. Best practices summary

Goal: Equip you with practical knowledge to validate LLM-based text annotations and make informed deployment decisions.

Validation Framework

- Human Validation
- Labeled Data Validation
- External Proxy Validation

Running LLMs for Annotation

- Deployment Options
- Prompting Strategies
- Privacy and Guardrails

Best Practices Summary

Why Validate LLM Annotations?

The Promise

- Annotate large-scale at low cost
- Consistent application of coding rules
- Handles nuance and complexity better than keywords (and other methods)
- Faster iteration than human coding

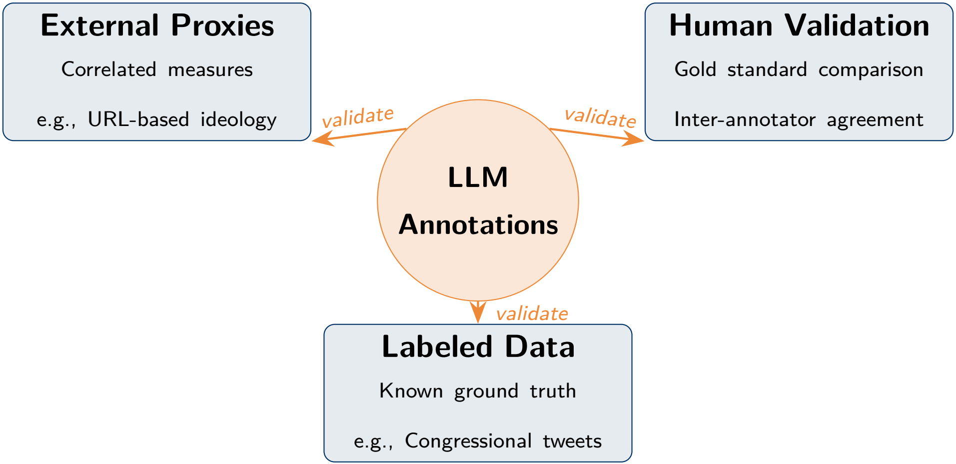
The Risk

- Black-box
- Unknown error rates without validation
- Systematic biases from training
- Hallucination, overconfidence
- Full replicability not guaranteed
- Model updates

Take-away

→ Validation is not optional – we want **measurement**, not **guessing**. Ideally, multiple validation approaches provide complementary evidence of reliability.

The Validation Triangle



Running Example: Political Twitter/X Study

Research Context

Study of user behavior and political content on Twitter/X

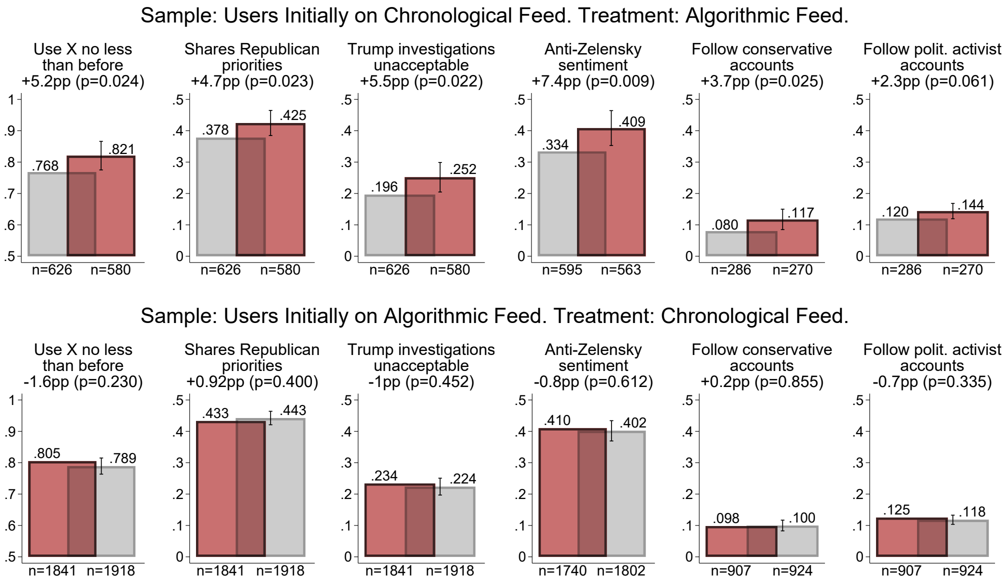
Research Questions

- Political leaning of content/accounts?
- What account types dominate?
- How do methods compare?

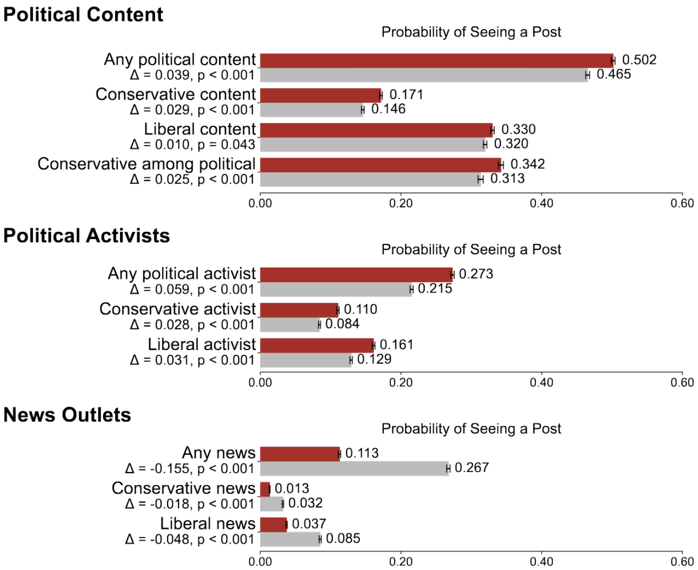
Annotation Challenge

- Millions of accounts – manual annotation infeasible

Running Example: Political Twitter/X Study



Running Example: Political Twitter/X Study



Dataset Overview

Data Collection (Text)

- **Platform:** Twitter/X
- **Period:** Summer '23
- **Scope:** Feed samples, followed accounts
- **Unit:** Account-level analysis

Validation Data Sources

Source	Description
Human coders	4 annotators, 500 accounts each
Congress tweets	353,742 tweets from 968 members
URL ideology	Domain-level political slant scores

Multiple independent data sources enable robust validation of annotations.

Case Study: LLM Annotation Setup

Model Configuration

- **Model:** Llama 3.3 70B Instruct
- **Temperature:** 0 (improves reproducibility but is not perfectly deterministic)
- **Input:** Bio (+ sample of recent posts)

Annotation Dimensions

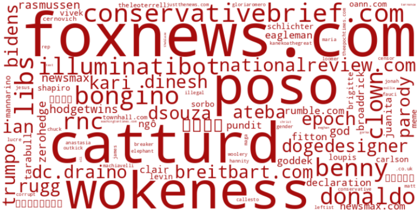
1. **Political Leaning**
   - Conservative / Liberal / Cannot say
2. **Content Type**
   - News / Political activist / Entertainment / Official / Other

Prompt Structure

```
I will show you the name, description, and
tweets from a Twitter account.
Classify the account's political leaning.
Labels: Conservative, Liberal, Cannot say
Account name: [...]
Description: [...]
Sample tweets: [...]
```

LLM Annotation Results: Word Patterns by Category

Conservative Accounts



Liberal Accounts

Wordclouds reveal distinctive vocabulary patterns that LLMs leverage for classification.

Human Validation: Methodology

Study Design

- **Annotators:** 4 US-based human coders
- **Sample:** 500 accounts per annotator
- **Task:** Same dimensions as LLM
- **Overlap:** Subset coded by multiple annotators

Key Metrics

- **Inter-annotator reliability:** Krippendorff's α
- **LLM vs. human agreement:** Confusion matrix
- **Performance:** Precision, Recall, F1-score

What is Krippendorff's Alpha?

Definition

Measures agreement among annotators, accounting for chance.

$$\alpha = 1 - \frac{D_o}{D_e}$$

D_o = observed disagreement; D_e = expected

Why use it?

- Works with any number of annotators
- Handles missing data
- Corrects for chance agreement

Interpretation Scale

α	**Interpretation**
> 0.80	Excellent
$0.67 - 0.80$	Good
$0.40 - 0.67$	Moderate
< 0.40	Poor

Key insight: If humans disagree, LLMs cannot achieve perfect accuracy.

Human Validation: Inter-Annotator Agreement

Krippendorff's Alpha Results

Dimension	α	**Interpr.**
Political Leaning	0.69	Good
Content Type	0.49	Moderate

Interpretation

- $\alpha > 0.80$: Excellent
- $\alpha > 0.67$: Good
- $\alpha > 0.40$: Moderate

Key Finding: Political leaning shows good agreement ($\alpha = 0.69$), content type is more ambiguous ($\alpha = 0.49$).

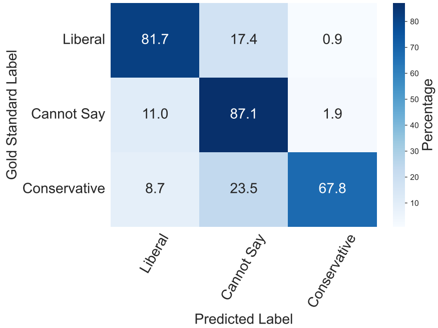
Why the difference?

- Political leaning: Clearer signals
- Content type: Subjective boundaries

This sets the ceiling for LLM accuracy!

Human Validation: LLM vs. Human Agreement

Political Leaning Confusion Matrix



Performance Metrics

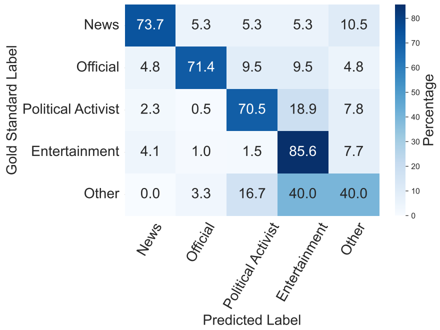
Class	P	R	F1
Liberal	0.76	0.93	0.84
Cannot say	0.78	0.74	0.76
Conservative	0.94	0.68	0.79
Macro avg	0.83	0.78	**0.80**

P = Precision, R = Recall

Overall: ~80% accuracy, comparable to human agreement

Human Validation: Content Type Performance

Content Type Confusion Matrix



Key Observations

- Overall accuracy: ∼75%
- News: Easiest to classify
- Entertainment vs. Other: Most confusion
- Reflects human disagreement

Takeaway

LLM performance tracks human agreement – harder for humans means harder for LLMs.

Validation with Known Ground Truth

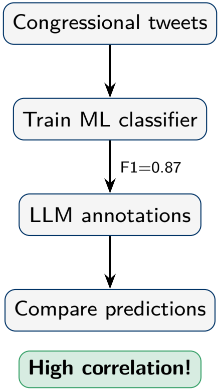
The Idea

- Use datasets with *known* labels
- Compare LLM predictions to ground truth
- No human annotation needed

Congressional Tweets Dataset

- **Source:** Congress member accounts
- **Size:** 353K tweets, 968 members
- **Labels:** Party affiliation (R/D)

Party affiliation is **objective ground truth**!



ML Classifier Approach

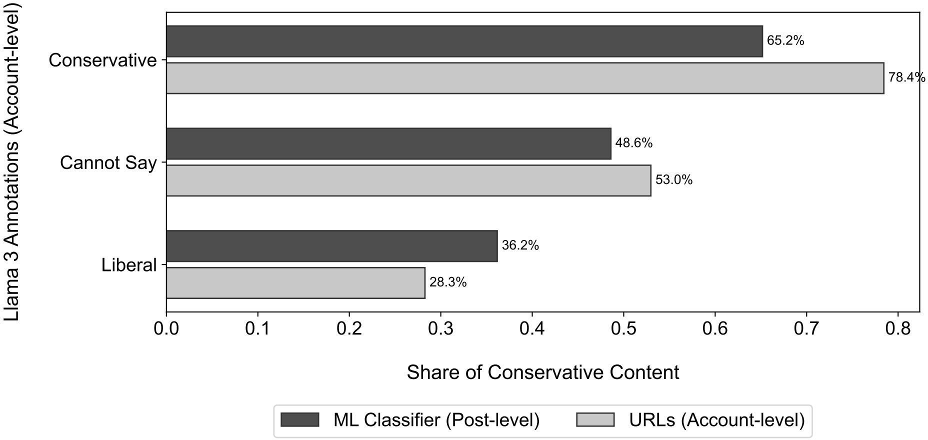
Methodology

1. Train classifier on Congressional tweets
   - Word frequency features
   - Known party labels
2. Apply to general accounts
3. Compare with LLM predictions

Classifier Performance

- F1-score on test set: **0.87**
- Strong generalization to political language

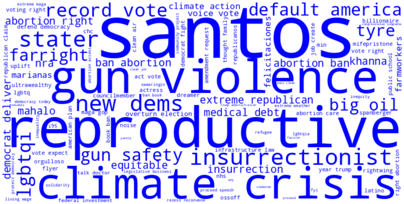
ML Classifier Approach



ML Classifier: Distinctive Language Patterns

Republican Congress Members

Democratic Congress Members



Word frequencies from Congressional tweets form the basis for the ML classifier's predictions.

External Proxy Validation: URLs

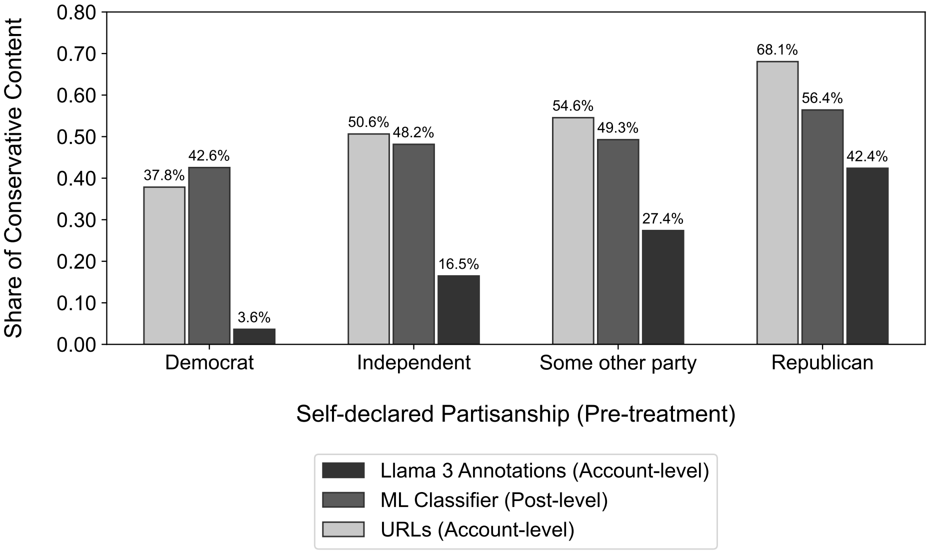
URL-Based Ideology Measures

- Shared URLs have known ideological slant
- Example: Breitbart (cons.), MSNBC (lib.)
- Aggregate URL sharing patterns per account
- Correlate with LLM political annotations
- Selective: only if URL present

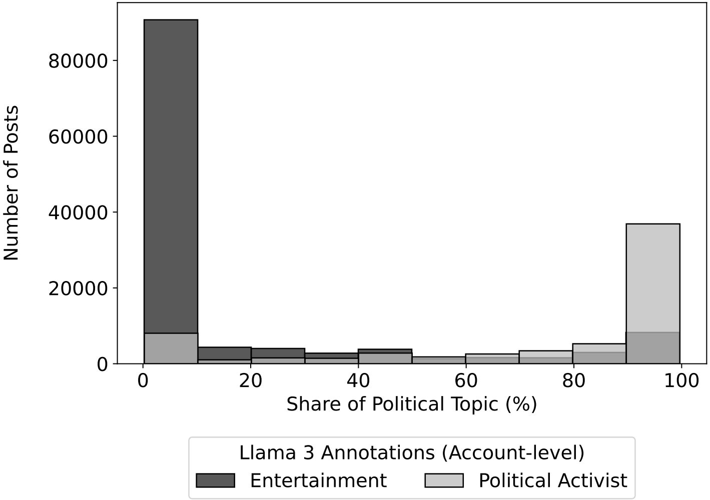
Other Potential Proxies

- Topics
- Hashtags

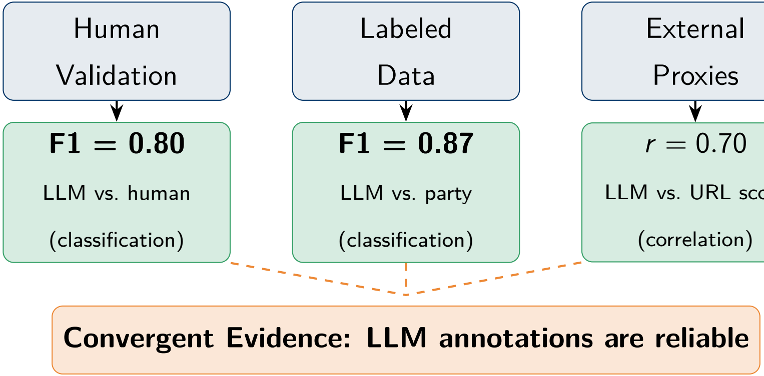
External Proxy Validation: URLs



External Proxy Validation: Topics



Validation Summary: Convergent Evidence



Why different metrics? F1-score for categorical comparisons (LLM labels vs. categorical ground truth). Correlation (r) for continuous comparisons (LLM labels vs. continuous ideology scores from URL sharing patterns).

API vs. Local Deployment

API-Based

- \+ Easy setup, no hardware
- \+ Access to latest models
- \+ Automatic scaling

- – Usage costs add up
- – Data leaves your server
- – Limited fine-tuning
- – Rate limits apply

Providers: OpenAI, Anthropic, Google, Groq, Together.ai

Local/Self-Hosted

- \+ Full data control
- \+ No per-query costs
- \+ Full fine-tuning
- \+ No rate limits

- – GPU hardware needed
- – Setup complexity
- – Maintenance burden
- – Limited to smaller models

Tools: Ollama, vLLM, llama.cpp, HuggingFace

API Options for Research

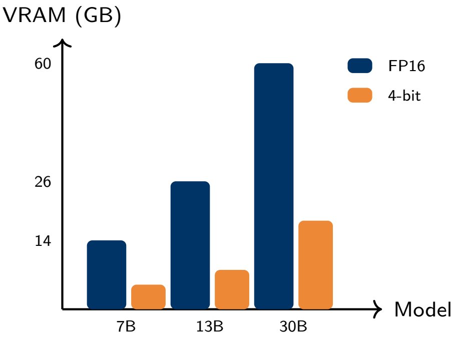
Provider	Key Models	Fine-tuning	Cost	Notes
OpenAI	GPT-4o, GPT-4o-mini	Yes (limited)	$$$	Most popular
Anthropic	Claude 3.5 Sonnet	No	$$$	Strong reasoning
Google	Gemini 1.5 Pro	Yes	$$	Long context
Groq	Llama 3.x, Mixtral	No	$	Very fast
Together.ai	Open-source models	Yes	$$	Flexible

A personally recommended budget option for non-sensitive data: Groq offers fast inference for open-source models at low cost.

Local Deployment: Hardware Requirements

GPU Memory Requirements

Model Size	Full (FP16)	Quantized (4-bit)
7B params	14 GB	4–5 GB
13B params	26 GB	8 GB
30B params	60 GB	18 GB
70B params	140 GB	35 GB



Common GPU Options

- RTX 4090: 24 GB (∼$2,000)
- A100: 40/80 GB (cloud: $2–4/hr)
- Consumer: RTX 3090 (24 GB)

Quantization enables running larger models on consumer hardware.

Quantization: Running Large Models Locally

What is Quantization?

- Reduce precision of model weights
- FP32 $\rightarrow$ FP16 $\rightarrow$ INT8 $\rightarrow$ INT4
- Trades accuracy for memory/speed

Quality: 8-bit ($<$1% loss), 4-bit ($\sim$2–5% loss)

Temperature and Reproducibility

What Temperature Controls

- **Temperature = 0:** Greedy decoding
- **Temperature > 0:** Adds randomness
- For annotation: always use temp = 0

Important Caveat

Temperature = 0 is **not perfectly deterministic**:

- Floating-point variations
- GPU parallelism
- Server load balancing

Provider Documentation

OpenAI	"Mostly deterministic"
Anthropic	"Not fully deterministic"
Google	Seed is "best-effort"

Best practice: Run multiple passes, report stability, document model version.

Prompting Strategies Overview

+ examples + training

Zero-shot	Few-shot	Fine-tuning
No examples	3–10 examples	100s–1000s examples
Just instructions	In-context learning	Model weight updates
Effort: Low	Effort: Medium	Effort: High
Quality: Variable	Quality: Good	Quality: Best
Start here	*Experiment*	*If needed*

Recommendation: Start with zero-shot, experiment with few-shot examples, consider fine-tuning based on performance needs and annotation effort.

Zero-Shot Prompting

Structure

1. Clear task definition
2. Output format specification
3. Classification categories
4. The text to classify

Best Practices

- Be explicit about categories
- Specify format (JSON, single word)
- Include "Cannot determine" option
- Set temperature $= 0$ for consistency

Example Prompt

```
Classify the political leaning of this Twitter
account based on their bio and recent posts.

Categories:
- Conservative
- Liberal
- Cannot determine

Output only the category name.

Bio: [account bio]
Posts: [sample posts]
```

Few-Shot Prompting

Adding Examples

- Include 3–10 labeled examples
- Cover all categories
- Include edge cases

Example Selection Tips

- Representative of each class
- Balance across categories

Caveat: Example selection can appear arbitrary (e.g., to referees) – document choices.

Few-Shot Template

```
[Task description]
Ex 1: "MAGA..." → Conservative
Ex 2: "Progressive..." → Liberal
Ex 3: "Cat lover..." → Cannot determine
Now classify: [new account]
```

Fine-Tuning: When and How

When to Fine-Tune

- Few-shot performance insufficient
- Domain-specific terminology
- Very large annotation volume
- Need for (better) reproducibility

Requirements

- Labeled training data (250–500+)
- Access to fine-tunable model
- Computational resources
- Validation set for evaluation

Fine-Tuning Options

Method	Notes
Full	Expensive, best results
LoRA	Efficient, popular
QLoRA	Memory-efficient
OpenAI	API fine-tuning
Together	Open models
Local	Full control

Note: Claude (Anthropic) does *not* support fine-tuning.

Privacy Considerations

Data Sensitivity Questions

- Does data contain PII?
- Is data subject to IRB approval?
- Can data leave institutional servers?

Training Defaults

- **API/Enterprise:** Not used for training
- **Consumer:** Used by default; opt-out available

Privacy-Preserving Options

Locally deployed	Data stays local
Anonymization	Remove PII first
Enterprise APIs	No training use

Rule: When in doubt, use local deployment or consult IRB.

Guardrails and Content Restrictions

The Problem

- LLMs may refuse sensitive content
- Extremist, violent, or sexual content
- Inconsistent refusal patterns

Research Implications

- Cannot annotate certain content via API
- Refusals create missing data, bias toward "safe" content

Solutions

- **Local models** – finetuning possible
- **System prompts** – research context
- **Researcher access** – provider programs
- **Pre-filtering** – remove extreme content

Best Practices Summary

Validation

1. Always validate – never assume accuracy
2. Use multiple validation methods
3. Report inter-annotator agreement
4. Compare to human performance ceiling

Reproducibility

5. Set temperature $= 0$ (not fully deterministic)
6. Document model version and date
7. Share prompts and code

Implementation

8. Start with zero or few-shot prompting
9. Pilot test on small sample
10. Build error analysis into workflow
11. Consider privacy early

Reporting

12. Report precision, recall, F1
13. Show confusion matrices
14. Discuss limitations

Remember: It's still supervised learning!

Key principle: LLMs are just another supervised learning approach

All the usual rules apply (see Lecture 3):

- **Train/validation/test split**: Don't evaluate on training data!
- **Class imbalance**: Handle appropriately
- **Overfitting**: Monitor validation performance
- **Metrics**: Choose appropriate for your task (accuracy, F1, etc.)

The current research frontier

- Researchers increasingly generate key variables (labels, scores, embeddings, latent constructs) using LLMs / ML, and then use them in regressions.
- Treating these generated quantities as observed data can create bias and invalid inference due to prediction/measurement error.
- Main references of this (still very recent) literature: Egami, Hinck, Brandon M. Stewart, et al. 2023; Egami, Hinck, Brandon M Stewart, et al. 2024; Battaglia et al. 2024; Ludwig et al. 2024; Carlson and Dell 2025

Questions?

Thank you for your attention

Germain Gauthier, Philine Widmer

Bocconi University, Paris School of Economics

References I

Battaglia, Laura et al. (2024). *Inference for Regression with Variables Generated by AI or Machine Learning*. Revised Apr 2025. arXiv: 2402.15585.

Carlson, Jacob and Melissa Dell (2025). "A Unifying Framework for Robust and Efficient Inference with Unstructured Data". In: *arXiv preprint arXiv:2505.00282*.

Egami, Naoki, Musashi Hinck, Brandon M Stewart, et al. (2024). "Using Large Language Model Annotations for the Social Sciences: A General Framework of Using Predicted Variables in Downstream Analyses". In: *Preprint from November 17*, p. 2024.

— (2023). "Using Imperfect Surrogates for Downstream Inference: Design-based Supervised Learning for Social Science Applications of Large Language Models". In: *Advances in Neural Information Processing Systems* 36.

References II

Ludwig, Jens, Sendhil Mullainathan, and Stefan Rambachan (2024). *Large Language Models: An Applied Econometric Framework*. Working Paper 33344. National Bureau of Economic Research.