Data Science: Research Systems and Discoveries

Mike Cafarella

Part 1: Systems

Foundation Models are Great!

- LLMs, VLMs, OpenCLIP, other models are incredible potential building blocks
- Consider the *vast range* of possible data/AI applications

Data Integration	**Next-Generation Search**
Data Cleaning	**Next-Generation Dashboards**
Information Extraction	**Log-Driven System Diagnosis**
Form Processing	**Data-Driven Digital Twins**
Multimodal Scientific Discovery	**… and many others**

- All of these are small use cases today, but possibly huge tomorrow
- All of these should now be dramatically easier to build

…but AI Programming is a Drag

- The user has to complete a correct software engineering goal
    - “Find all the materials science papers that talk about EV batteries”
    - “Find all US banks’ SEC filings in 2022 and extract the footnotes that talk about solvency”
    - “Extract a video of the winning touchdown from every Super Bowl”

- **While also** ensuring good quality, fast execution, and reasonable costs

- **While also** models, hardware, and optimization methods are in constant flux

- **While also** projects needs change over time (e.g., minimal cost vs maximal quality)

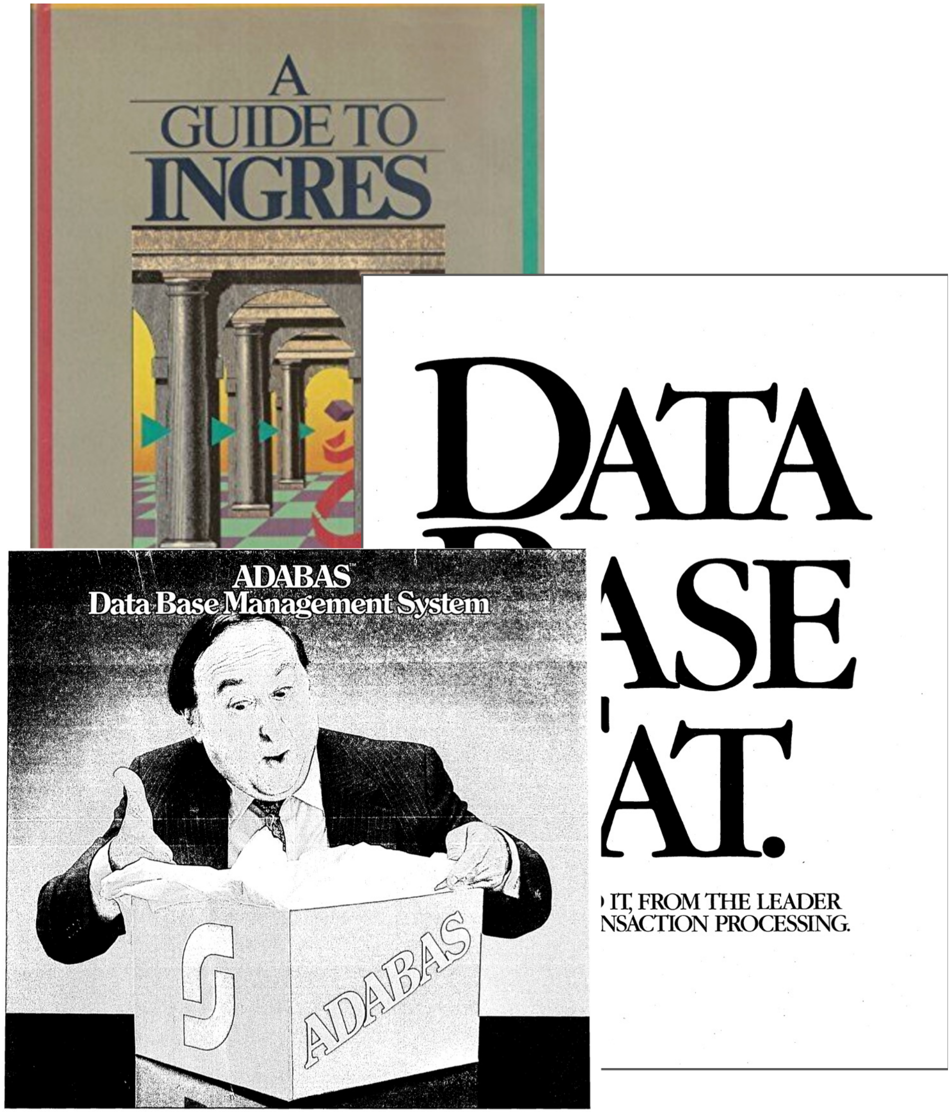
The Good News

We’ve solved a problem like this before!

In the mid-1970s, database programmers had to write custom code for every query

Declarative queries allowed them to write succinct programs while also obtaining good performance in a rapidly-changing technological environment

Let’s do the same for data/AI applications



Sample Code

“Get the author and subject of every email in the Enron collection”

```
class Email(pz.TextFile):
   """Represents an email, which in practice is usually from a text file"""
   sender = pz.Field(desc="The email address of the sender", required=True)
   subject = pz.Field(desc="The subject of the email", required=True)

emails = pz.Dataset(“enron-collection”, schema=Email)
```

Sample Code

“Find all the materials science papers about EV batteries that come from MIT and report all the paper metadata”

```
class ScientificPaper(pz.PDFFile):
  """Represents a scientific research paper, which in practice is usually from a PDF file"""
  title = pz.Field(desc="The title of the paper.", required=True)
  publicationYear = pz.Field(desc="The year the paper was published", required=False)
  author = pz.Field(desc="The name of the first author of the paper", required=True)
  institution = pz.Field(desc="The institution of the paper", required=True)

sciPapers = pz.Dataset(“materials-science-papers”, schema=ScientificPaper)
filteredPapers = scientificPapers.filterByStr(“The paper is about batteries”)
output = filteredPapers.filterByStr(“The paper is from MIT”)
```

Sample Code

“Find the images that contain at least one dog and figure out its breed”

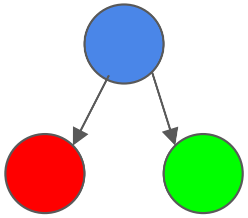
```
class DogImage(pz.ImageFile):
   breed = pz.Field(desc="The breed of the dog", required = True)

images = pz.Dataset(“image-corpus”, schema=pz.ImageFile)
filteredImages = images.filterByStr(“The image contains one or more dogs”)
dogImages = filteredImages.convert(DogImage, desc = “Image of a dog”)
```

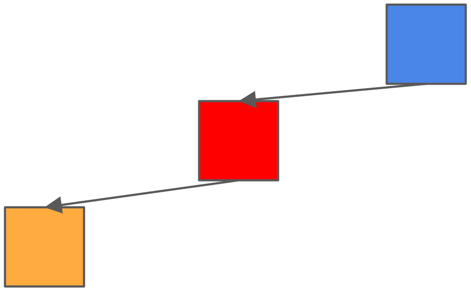
Execution

```
emails = pz.Dataset(“enron-collection”, schema=Email)
```

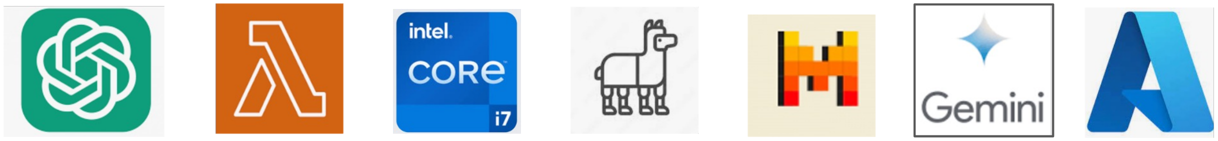
Step 1: User Query



Step 2: Logical Optimization



Step 3: Physical Optimization



Step 4: Concrete Execution

Optimizations are Crucial

These queries may process huge numbers of data objects.

Even minor parts of the query may naively entail multiple slow and expensive model invocations

LLM services deliver ~100 tokens/sec.
That’s less than 1kb/second

We need to automatically consider and choose many different optimizations to get better cost/quality tradeoffs

Choose cheap, lower-quality models when possible

Reduce input data size prior to LLM processing, if entire input isn’t needed

Synthesize traditional non-model code whenever possible

Approximate expensive LLM invocations with local one-off trained models

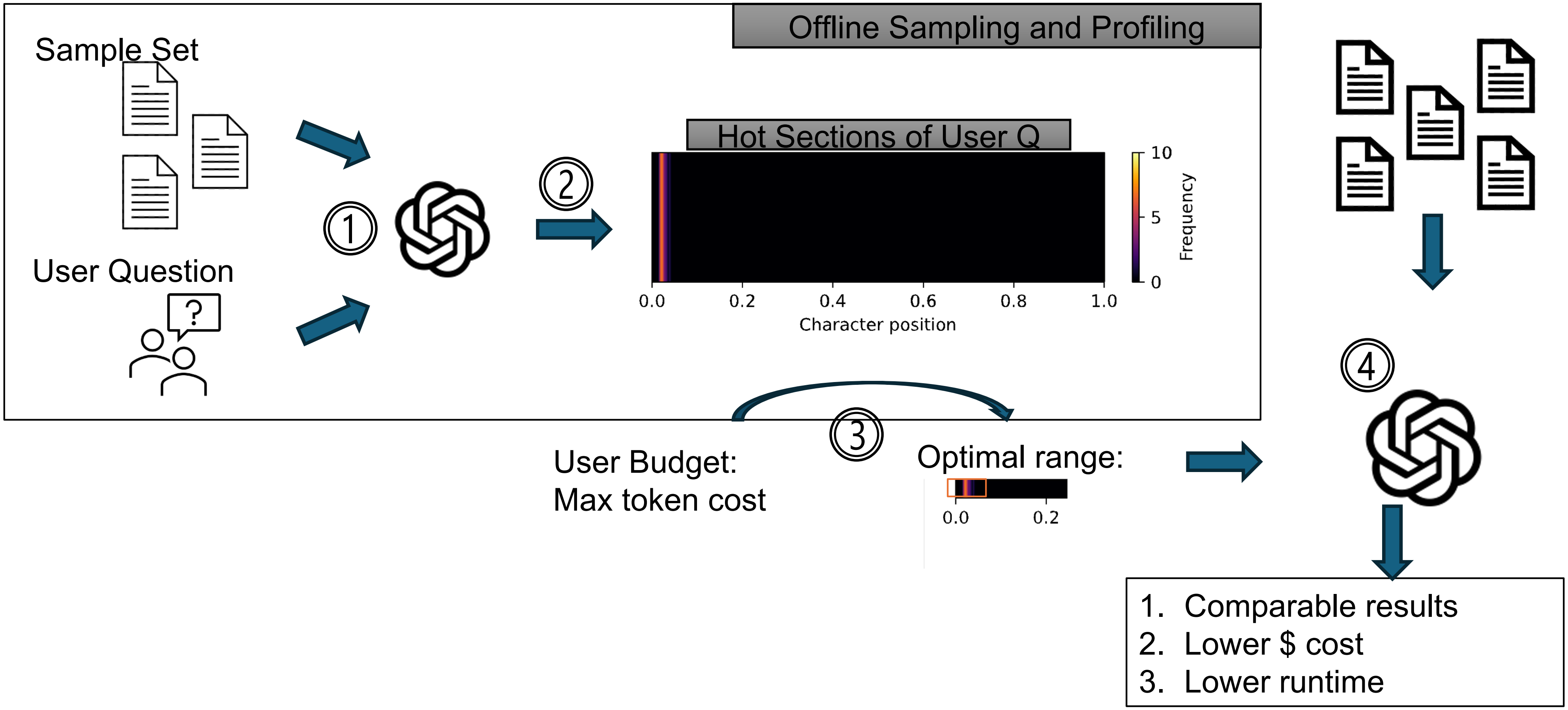
Use low-resolution imagery whenever possible

Use parallel execution services

… possibly many others?

Demo!

Sample-based Token Reduction



Offline Sampling and Profiling

Sample Set

User Question

Hot Sections of User Q

User Budget:
Max token cost

Optimal range:

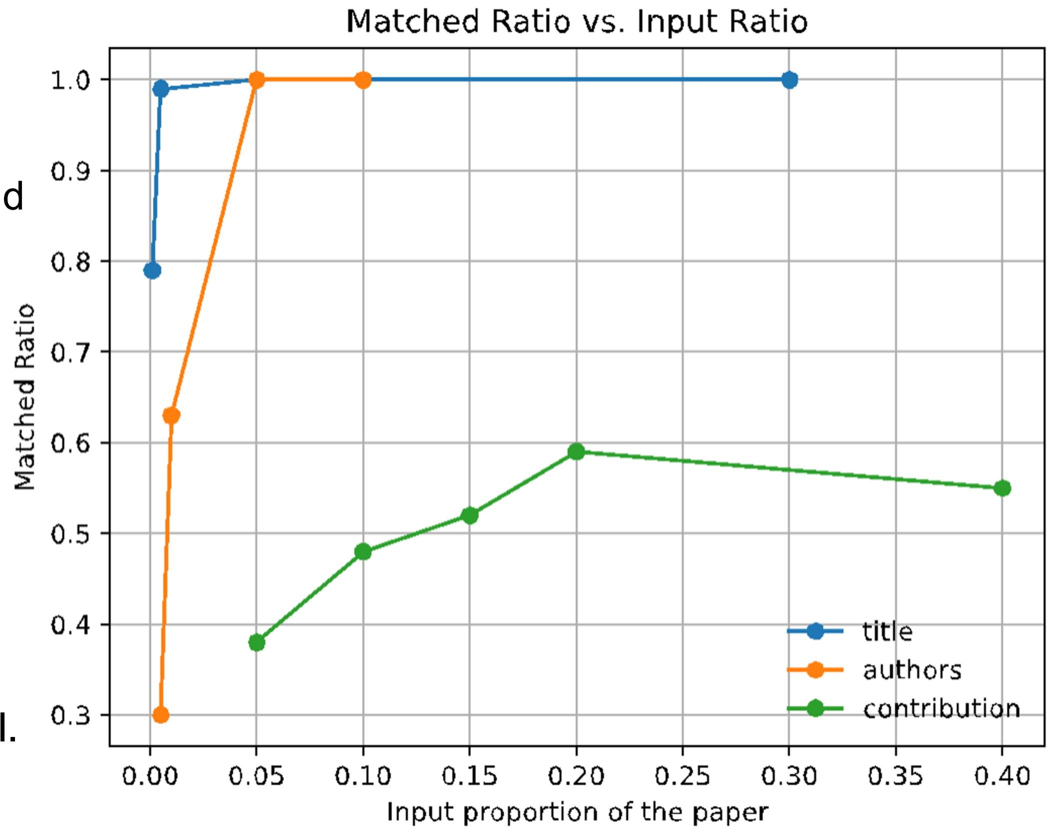
Test Set

1. Comparable results
2. Lower $ cost
3. Lower runtime

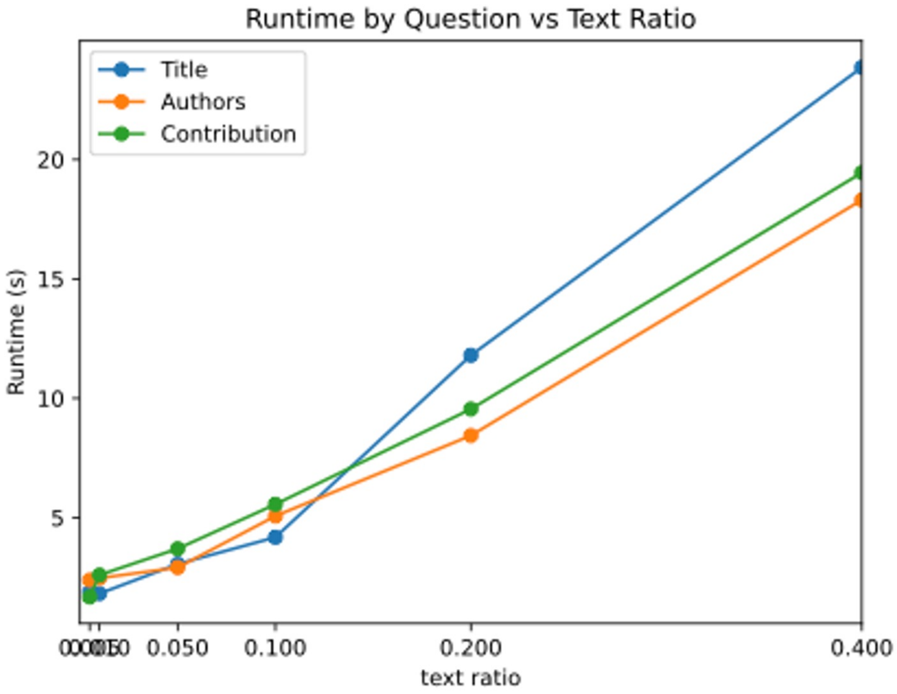
Comparable Results – Boolean Eval

Offline Experiment Setup:

- Get the heatmap over 80 VLDB papers and tested on another 100 VLDB papers
- QUESTIONS = [
"What is the main contribution of the paper?",
"Who are the authors of the paper?",
 "What is the paper title?"
]
- Varying input budget from 0.001-0.4 vs full.



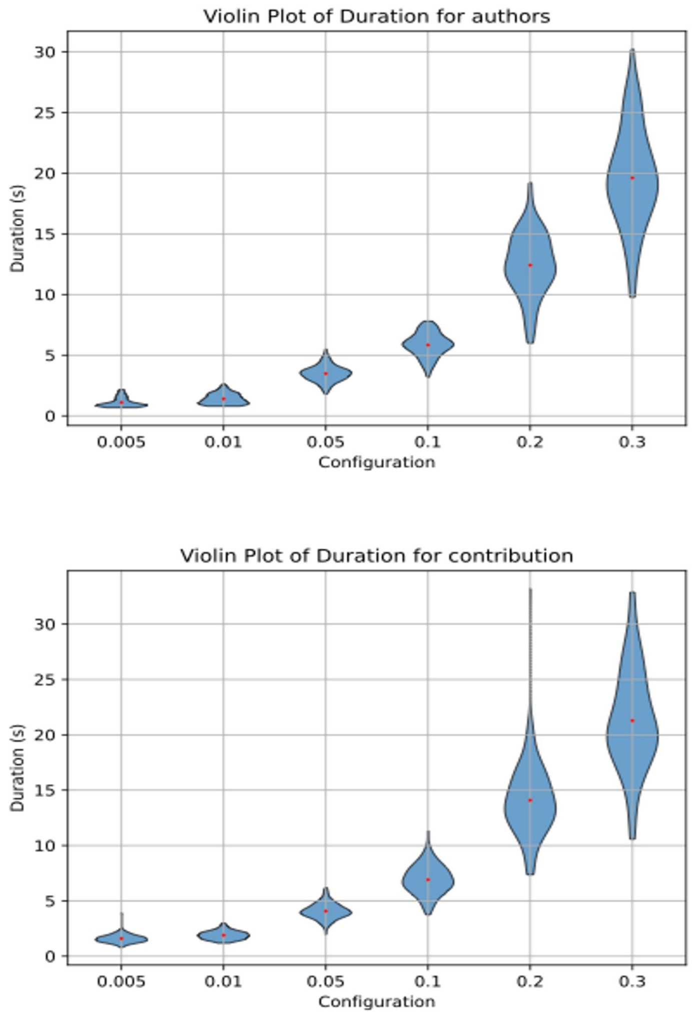
Inference Time (s)

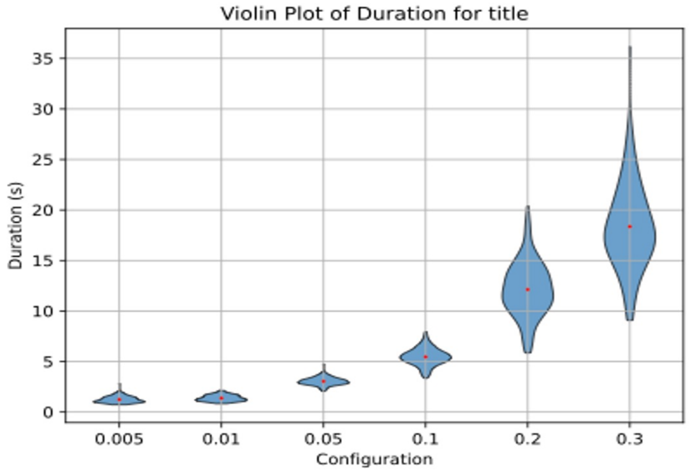


Text ratio→ Question↓	0.005	0.01	0.05	0.1	0.2	0.4 (~7k tokens)
Title	1.89	1.82	3.06	4.18	11.8	23.86
Authors	2.39	2.47	2.91	5.07	8.44	18.31
Contribution	1.68	2.60	3.70	5.56	9.55	19.45

Runtime with Huggingface API

- Runtime with Mistral-7B-Instruct-v0.2 on V100-32GB Memory





Convenience Features

- Cached answers are especially helpful, since many input sets don’t change
- Data marshalling and sampling is (or will be) built-in
- Improve quality with data labeling services & tools, without modifying the original code

Future Work

We have a working prototype, but this project is very large and we have a lot of work to do

- New optimization strategies
- New core data types: data plots, source code, videos, maps, blueprints, sequences and other bioinformatics data…
- Dynamic fine-tuning for improved optimization tradeoffs
- Cache answers across organizations and the internet
- New ancillary tools
- Streaming and improved performance

Part 2: Discoveries

Prices Continued to Rise in April, but Gains Slowed a Little: Live Updates

gor

ECONOMY | U.S. ECONOMY

U.S. March Jo

Momentum

Economists see labor m

along with a

a year earlier.

G.D.P. Report Shows th

Mask

MY

ation Accelerated to 8.5% in March, Hitting Four-Decade High

e index increase from year earlier driven by skyrocketing energy and food costs

How do we know?

Economists: Data Scientists Since Before It Was Cool

- Most modern federal statistical machinery grew up starting in 1930s
    - Statistical income research conducted in 1930s; modern income data series started in 1947
    - Consumer Price Index started in 1913
- Enabled by:
    - Legislation that compels survey response (e.g., Title 13 (Census Act))
    - Social norms around key voluntary surveys
        - Monthly retail sales; CPI enumeration
    - Statistical methods and research (e.g., Simon Kuznets)
    - Bureaucracy (Bureau of Labor Statistics, Census, BEA)

Inflation and Real Consumption Growth

- The price of a good changes: is that due to inflation or quality change?

2022: $100.00

2023: $90.00

+ $10.00

Inflation and Real Consumption Growth

- The price of a good changes: is that due to inflation or quality change?

2022: $100.00

2023: $110.00

- $10.00

Inflation and Real Consumption Growth

- The price of a good changes: is that due to inflation or quality change?

2022: $100.00

Now 20% Rayon

2023: $95.00

+/- $???.??

Quality Adjustment is Crucial

- Prices and quality vary simultaneously
  - Product quality varies in response to preferences, costs, etc
  - Collecting price data is relatively easy; quality adjustment is not

- Flux in market goods for sale is astonishing
  - 1-year half-life of a barcoded good (probability it will be on the shelf in 12 months) is about 50%

- Consider the vast number of products on the market (50K in supermarket)
- How can quality adjustment be done reliably, rigorously, affordably?
- Problem first practically examined in 1970s

Using ML to Construct Hedonic Price Indexes

Gabriel Ehrlich
Univ of Michigan

Tian Gao
Snowflake Inc

Matthew Shapiro
Univ of Michigan

John Haltiwanger
Univ of Maryland

Laura Yi Zhao
Bank of Canada,
Univ of Maryland

- Our plan:

1. Exploit large-scale product sales data from checkout scanners
2. Use machine learning to adjust for quality at barcode level
3. Employ resulting “well-behaved” dataset to compute new price index

Current BLS Adjust-for-Quality Algorithm

- Collect a set of (`time-period, item-id, price`) records

- By trade-weighted volume, about 15% of sales are adjusted using quantitative regression models (CPU speed, memory capacities, etc)

- Remaining sales are rule-adjusted

"house-to-house combat" approach (Shapiro and Wilcox 1996)

3. **Imputation**: adjust using average price change of product class

Modern Data Management Can Do Better

- Desiderata for better inflation and consumption data:
  - **Principled and accurate price adjustment**
  - High-frequency (as often as possible)
  - High-resolution (fine-grained product categories)

- Our plan:
  1. Exploit large-scale product sales data from checkout scanners
  2. Use machine learning to adjust for quality at barcode level
  3. Employ resulting “well-behaved” dataset to compute new price index

Price Data

- Nielsen transaction data
  - Weekly prices and quantities at store level
  - Supermarkets, groceries, discount, convenience, drug and liquor
  - Diverse set of goods
  - **Includes product descriptions**

'brand' ZR DT LN/LM CF NBP CT

'brand' NATURAL R CL NB 12P

'brand' DR W 1P 308S TT 6PK

Hedonic Adjustment at Scale

- Train a series of time-period-specific price prediction models
  - Training: 26.36M records, 2.2GB; takes ~88 hrs using GeForce

M_{2022}(“***brand*** ZR DT LN/LM CF NBP CT”) -> \$4.99

M_{2023}(“***brand*** ZR DT LN/LM CF NBP CT”) -> \$5.24

M_{2022}(“***brand*** DR W 1P 308S TT 6PK”) -> \$8.99

M_{2023}(“***brand*** DR W 1P 308S TT 6PK”) -> \$9.49

Our Procedure

1. Use appropriate model to predict prices for each time period
   - For new and disappearing goods, predicted price allows for contribution of new and exiting goods to inflation
   - For continuing goods, predicted price adjusts for changes in value of attributes
2. Compute aggregate inflation as weighted average of predicted price changes across all goods

Aggregating Price Adjustments

2022	2023
	 $95.00

Naive approaches:

- Mark goods as equal: **$5.00 price drop**
- Mark as incomparable: **$0.00 price change**

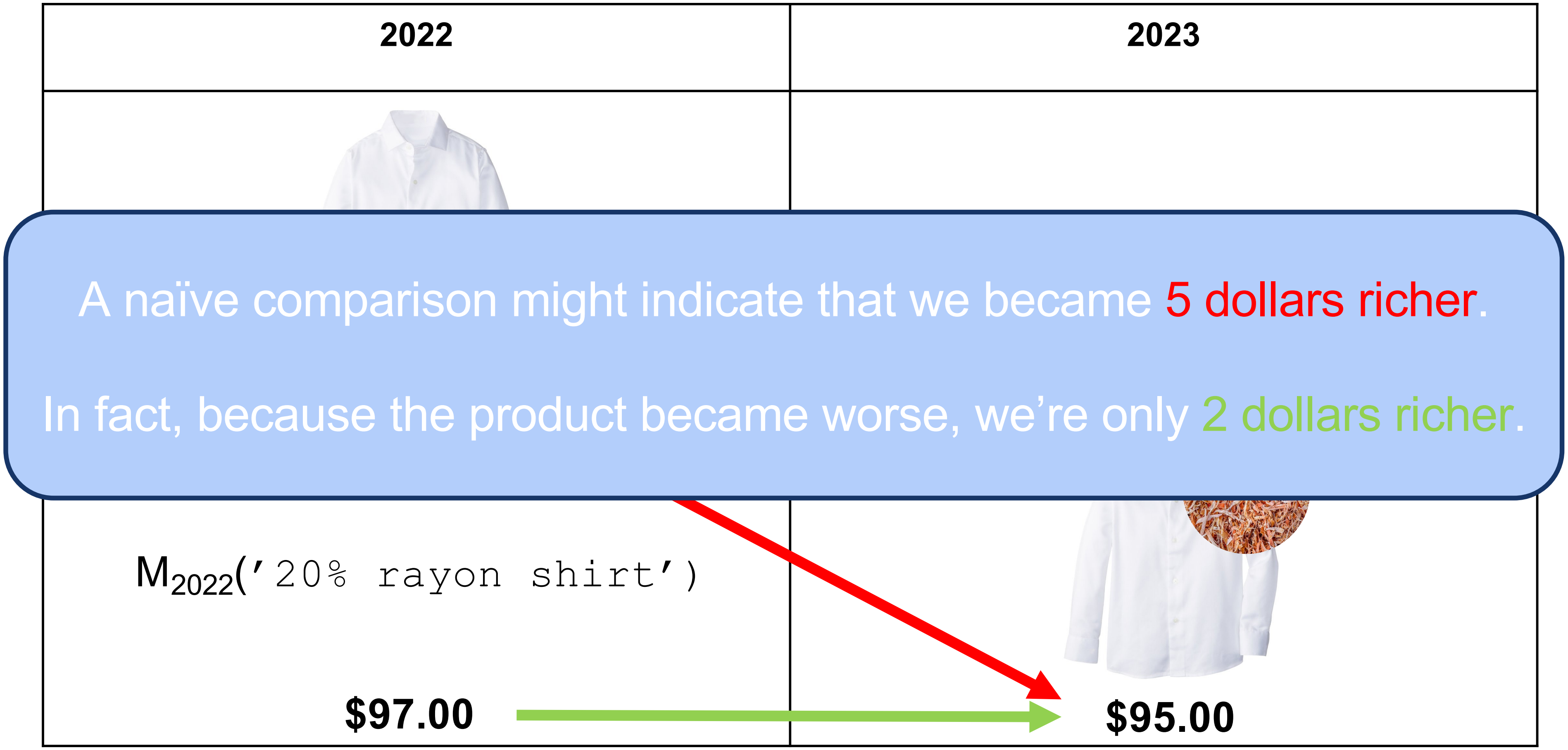
What benefit was delivered by a novel good?

VS

SMUCKER'S
GOOBER
Grape
Peanut Butter & Grape Jelly Stripes
NET WT 18 OZ (1 LB 2 OZ) 510g
AMERICA'S FAVORITE!

What benefit was lost by a good’s departure?

Aggregating Price Adjustments



2022	2023
M_{2022}(`'20% rayon shirt'`) **$97.00**	**$95.00**

A naïve comparison might indicate that we became 5 dollars richer.

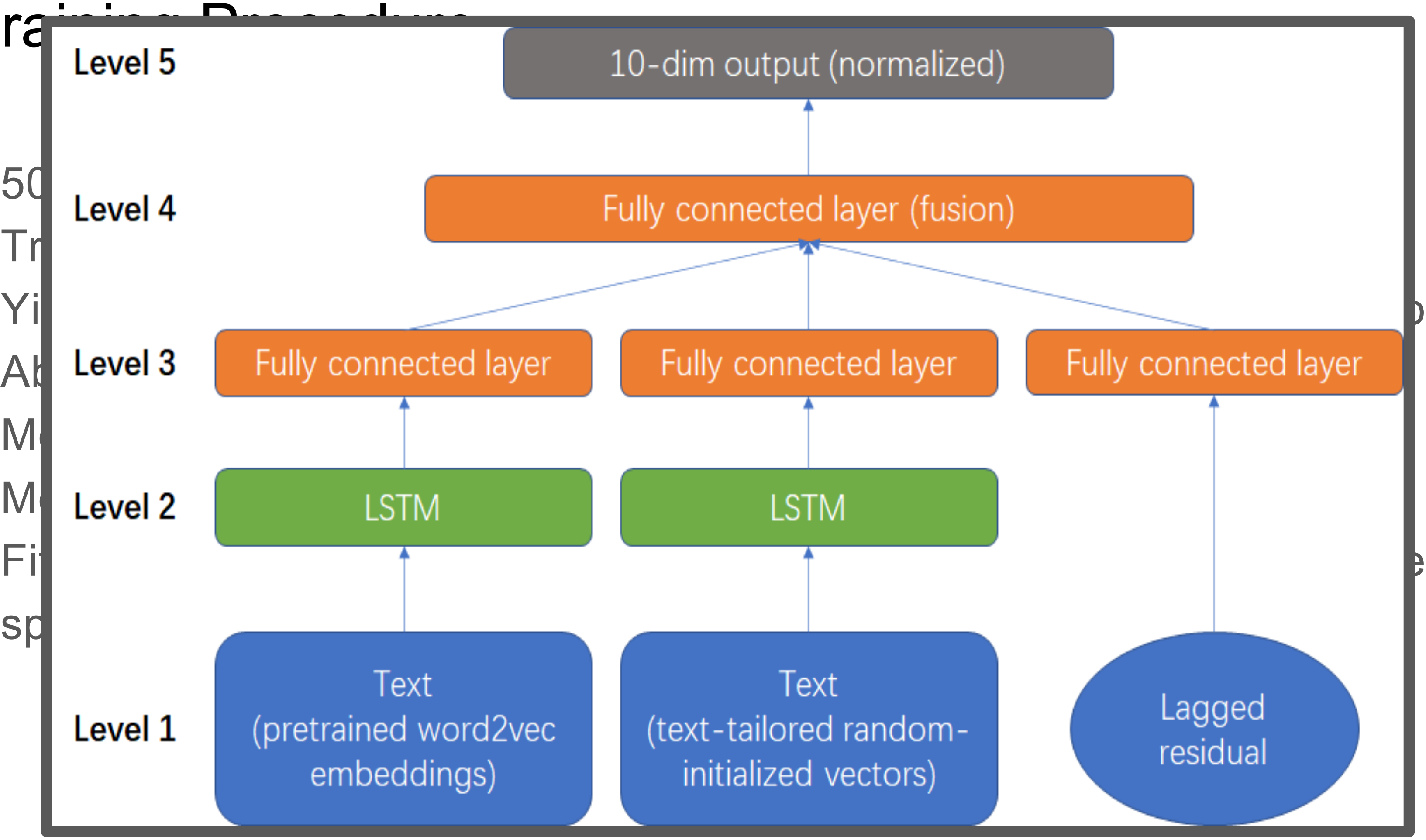
In fact, because the product became worse, we're only 2 dollars richer.

Training Procedure

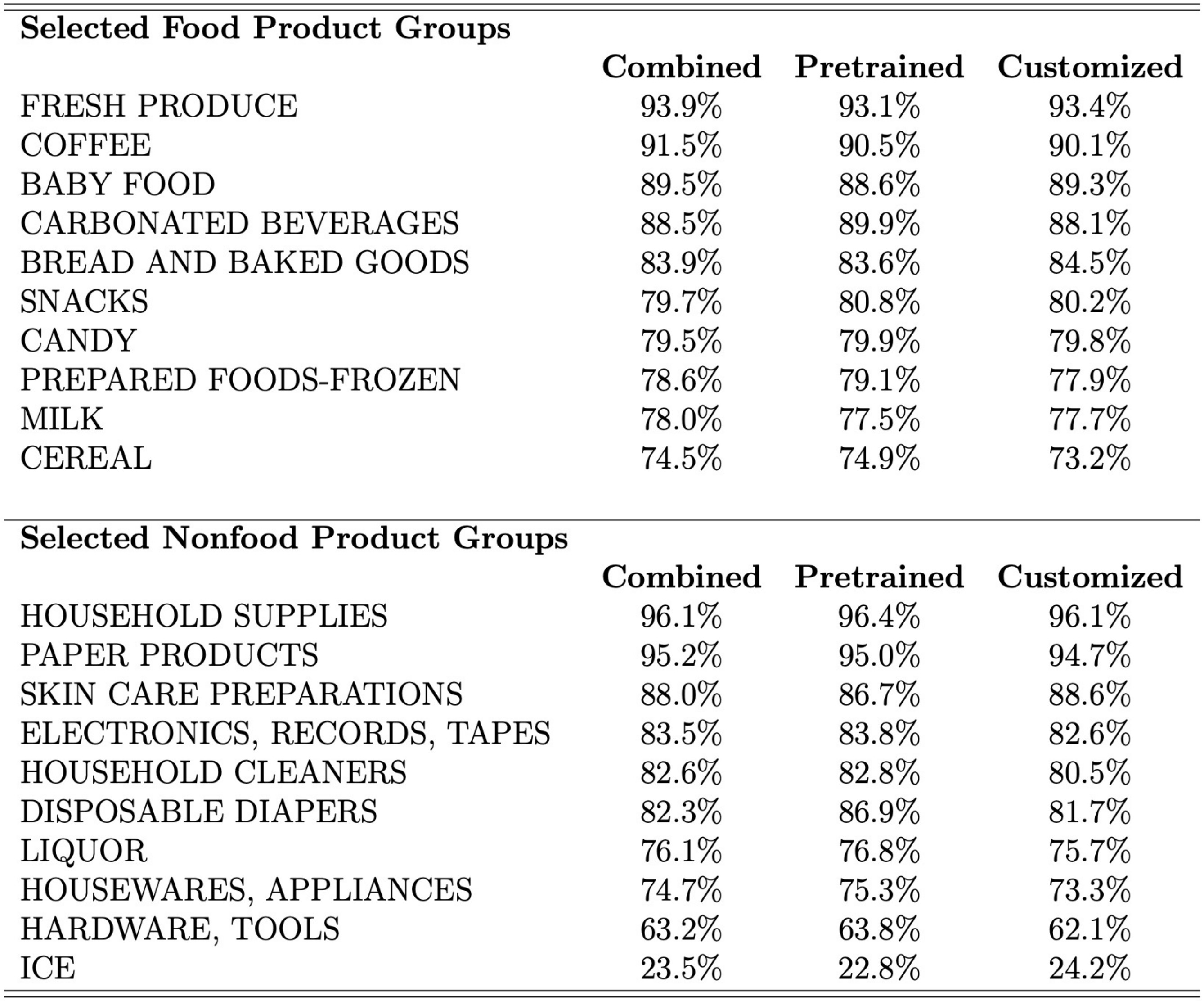
- 50/40/10 train/validate/test split
- Training data: 26.36M records, 2.2GB
- Yields 4,570 distinct models, one for each (year, quarter, product-group)
- About 88 hours of training time using NVIDIA GeForce
- Model implemented using PyTorch
- Model yields binned price prediction (10 deciles)
- Fit continuous prices as in product of non-trivial bin probability with time/product-specific bin means

Training Procedure

- 50
- Tr
- Yi)
- Ab
- M
- M
- Fi e/product-
  sp


Level 5
10-dim output (normalized)
Level 4
Fully connected layer (fusion)
Level 3
Fully connected layer
Fully connected layer
Fully connected layer
Level 2
LSTM
LSTM
Level 1
Text (pretrained word2vec embeddings)
Text (text-tailored random-initialized vectors)
Lagged residual

Model



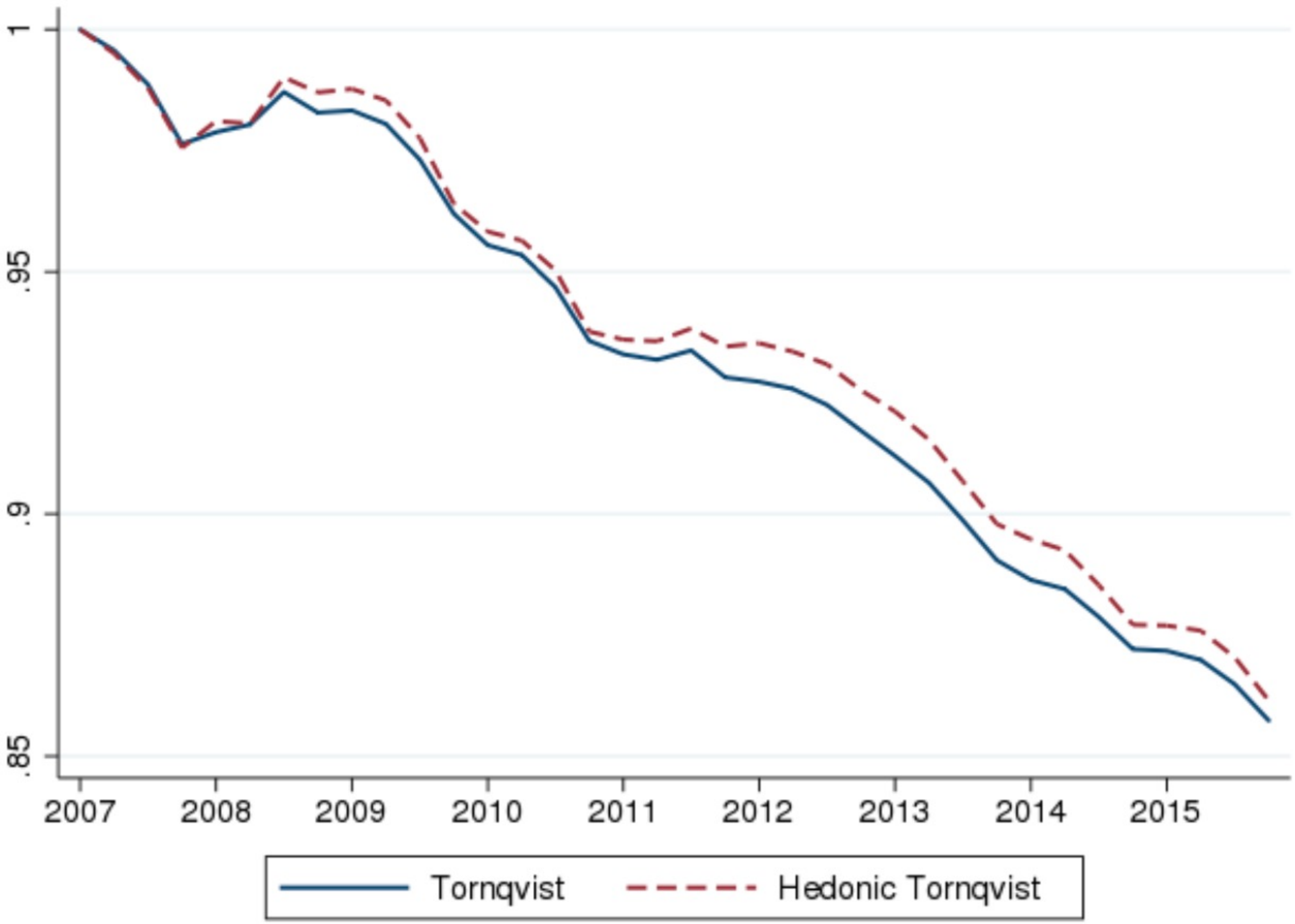
Selected Food Product Groups

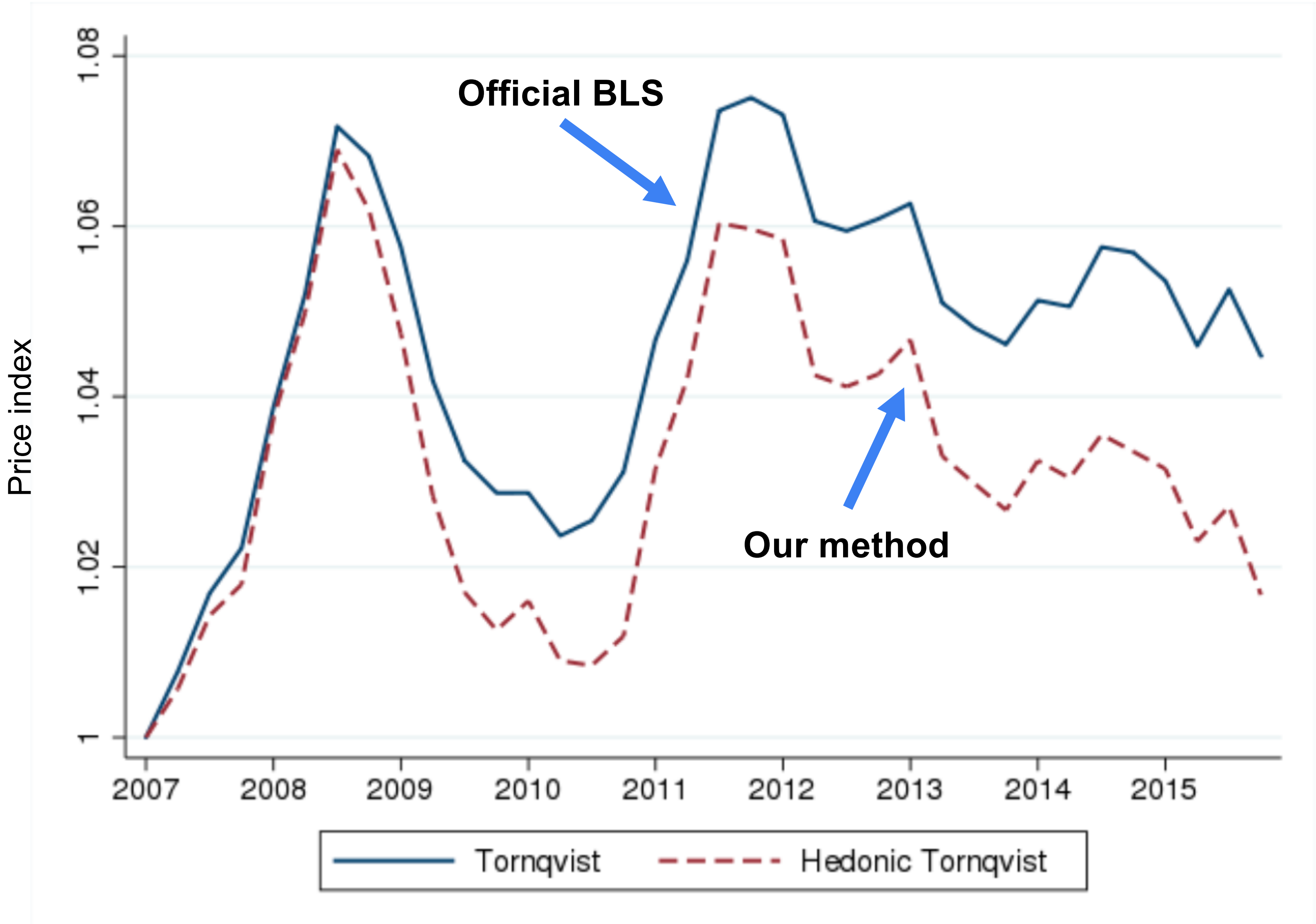
	Combined	**Pretrained**	**Customized**
FRESH PRODUCE	93.9%	93.1%	93.4%
COFFEE	91.5%	90.5%	90.1%
BABY FOOD	89.5%	88.6%	89.3%
CARBONATED BEVERAGES	88.5%	89.9%	88.1%
BREAD AND BAKED GOODS	83.9%	83.6%	84.5%
SNACKS	79.7%	80.8%	80.2%
CANDY	79.5%	79.9%	79.8%
PREPARED FOODS-FROZEN	78.6%	79.1%	77.9%
MILK	78.0%	77.5%	77.7%
CEREAL	74.5%	74.9%	73.2%

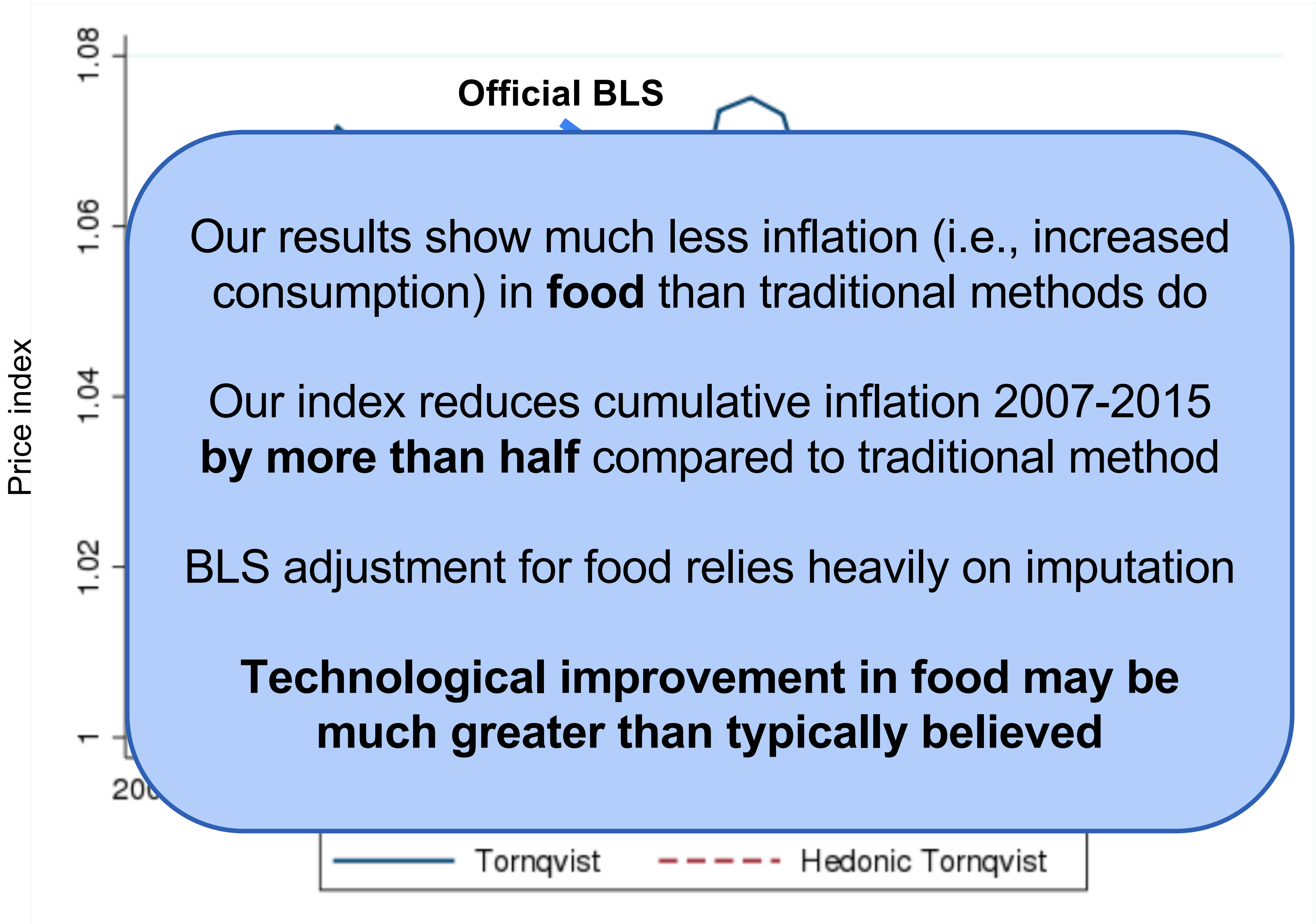
Selected Nonfood Product Groups

	Combined	**Pretrained**	**Customized**
HOUSEHOLD SUPPLIES	96.1%	96.4%	96.1%
PAPER PRODUCTS	95.2%	95.0%	94.7%
SKIN CARE PREPARATIONS	88.0%	86.7%	88.6%
ELECTRONICS, RECORDS, TAPES	83.5%	83.8%	82.6%
HOUSEHOLD CLEANERS	82.6%	82.8%	80.5%
DISPOSABLE DIAPERS	82.3%	86.9%	81.7%
LIQUOR	76.1%	76.8%	75.7%
HOUSEWARES, APPLIANCES	74.7%	75.3%	73.3%
HARDWARE, TOOLS	63.2%	63.8%	62.1%
ICE	23.5%	22.8%	24.2%

N


1
.95
.9
.85
2007
2008
2009
2010
2011
2012
2013
2014
2015
Tornqvist
Hedonic Tornqvist


Official BLS
Our method
Price index
1.08
1.06
1.04
1.02
1
2007
2008
2009
2010
2011
2012
2013
2014
2015
Tornqvist
Hedonic Tornqvist



Our results show much less inflation (i.e., increased consumption) in **food** than traditional methods do

Our index reduces cumulative inflation 2007-2015 **by more than half** compared to traditional method

BLS adjustment for food relies heavily on imputation

Technological improvement in food may be much greater than typically believed