

BWARE: Morphing Compression for Data-centric ML Pipelines

VLDB 2026

Sebastian Baunsgaard, Matthias Boehm

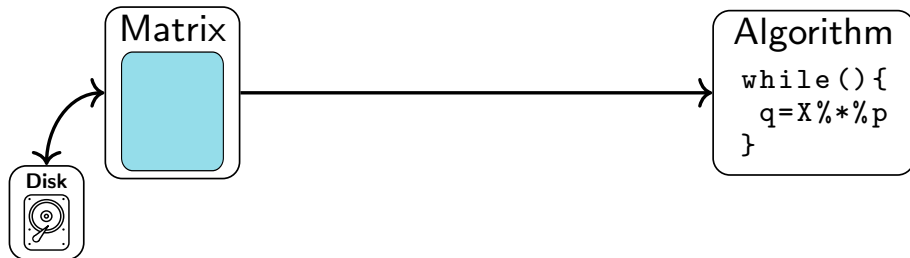
Overview



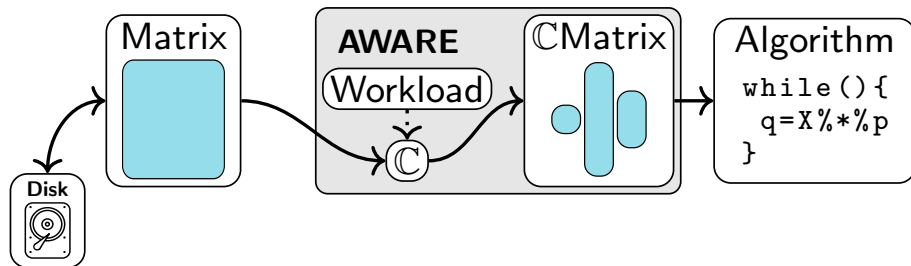
Algorithm

```
while() {  
  q=X%*%p  
}
```

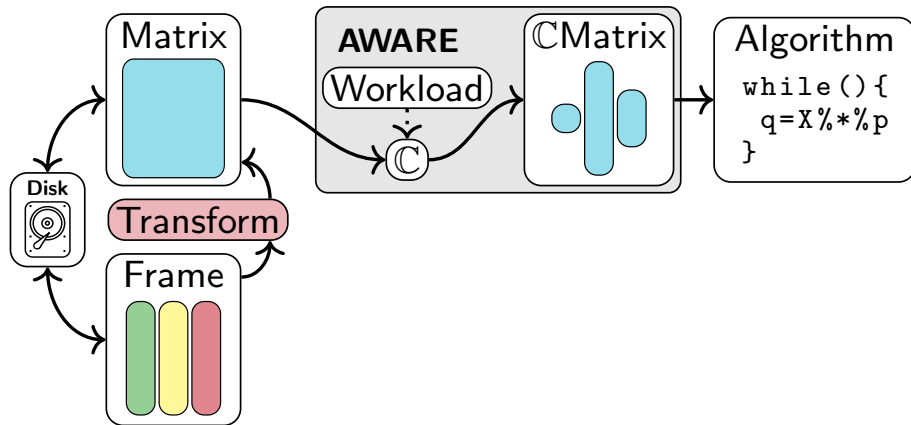
Overview



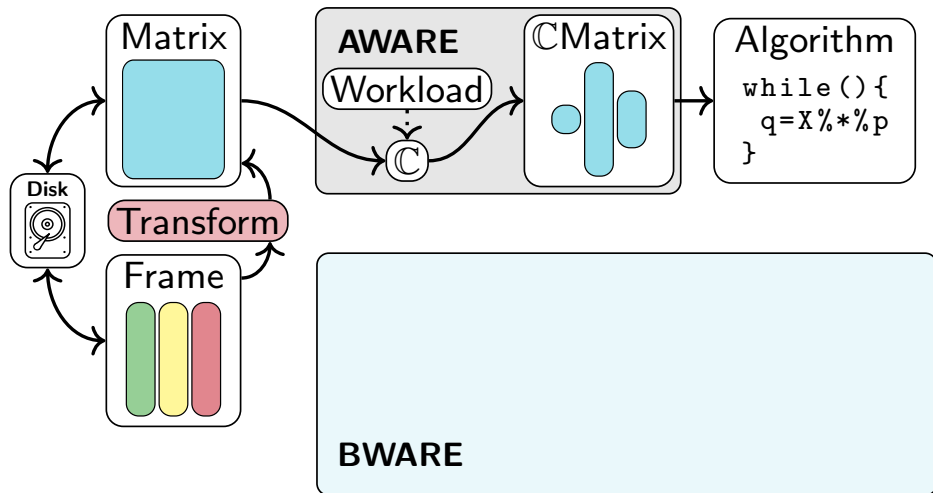
Overview



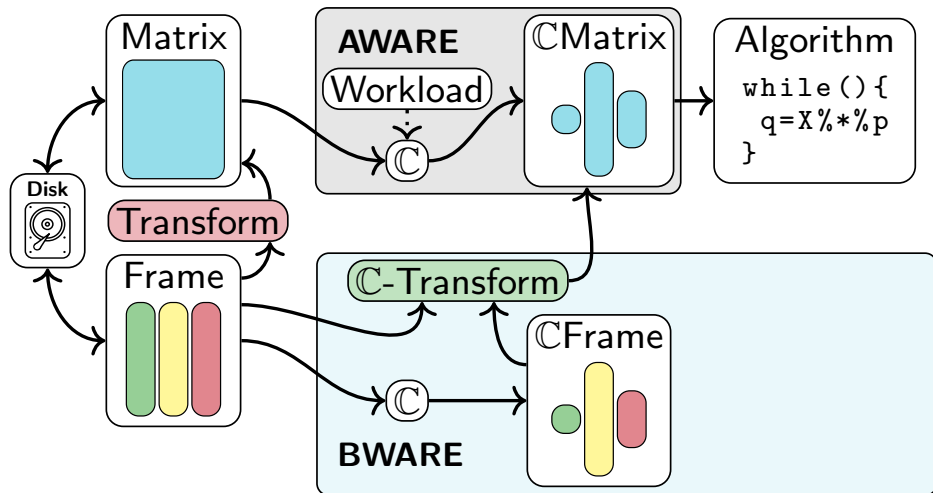
Overview



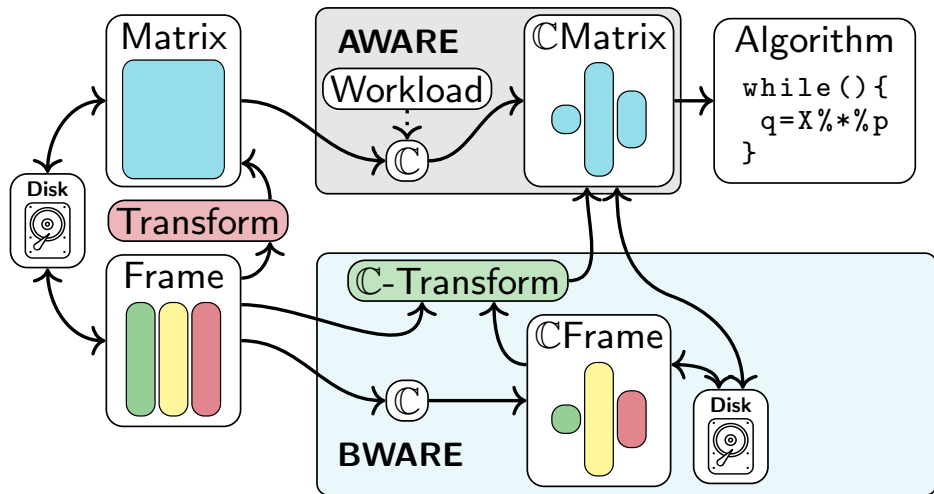
Overview



Overview



Overview



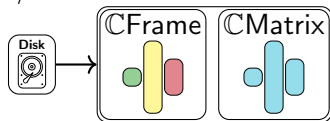




➤ BWARE: Morphing Compression for Data-centric ML Pipelines

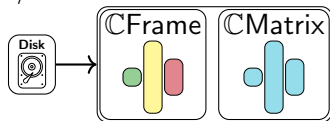
❖ BWARE: Morphing Compression for Data-centric ML Pipelines

❖ I/O

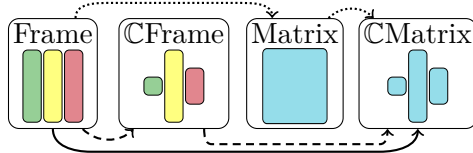


❖ BWARE: Morphing Compression for Data-centric ML Pipelines

❖ I/O

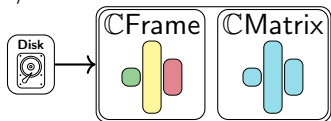


❖ Feature Transformation

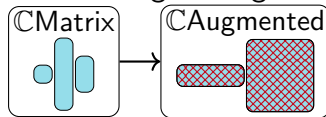


❖ BWARE: Morphing Compression for Data-centric ML Pipelines

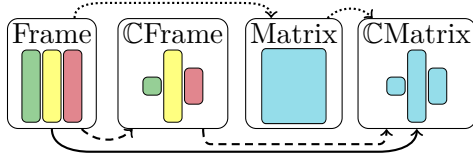
❖ I/O



❖ Feature Engineering

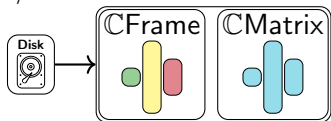


❖ Feature Transformation

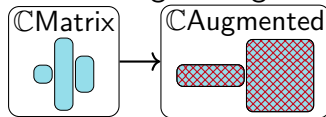


❖ BWARE: Morphing Compression for Data-centric ML Pipelines

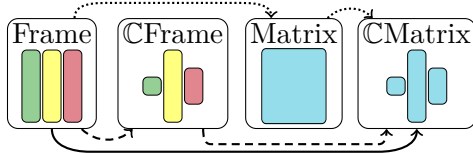
❖ I/O



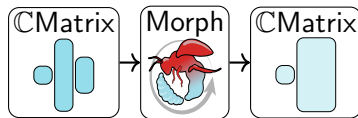
❖ Feature Engineering



❖ Feature Transformation

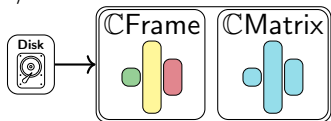


❖ Morphing

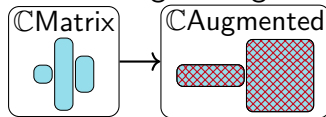


❖ BWARE: Morphing Compression for Data-centric ML Pipelines

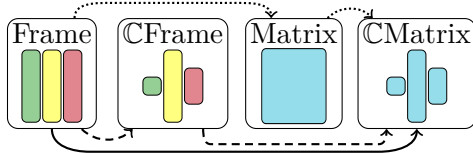
❖ I/O



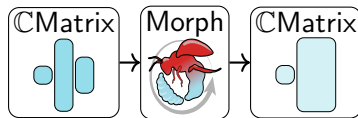
❖ Feature Engineering



❖ Feature Transformation



❖ Morphing



Avoid Rediscovering Compression Plans!



User Script:

```
X = read("data/X")  
y = read("data/y")  
  
X = scale(X, TRUE, TRUE)  
w = l2svm(X, y, TRUE,  
          1e-9, 1e-3, 100)  
  
write(w, "data/wXy")
```

User Script:

```
X = read("data/X")  
y = read("data/y")
```

```
X = scale(X, TRUE, TRUE)  
w = l2svm(X, y, TRUE,  
         1e-9, 1e-3, 100)
```

```
write(w, "data/wXy")
```

Built-in Functions:

```
if(shift)  
  X = X - colMeans(X)  
if(scale)  
  X = X / colSds(X)
```

```
if(intercept)  
  X = cbind(X, ones)  
while(conto & i<maxi) {  
  Xd = X %**% s  
  while(conti) {  
    out = 1-y*(Xw+sz*Xd)  
    sz = sz - g/h; # ...  
  }  
  g_new = t(X) %**% (out*y)  
}
```

Workload Awareness



User Script:

```
X = read("data/X")  
y = read("data/y")
```

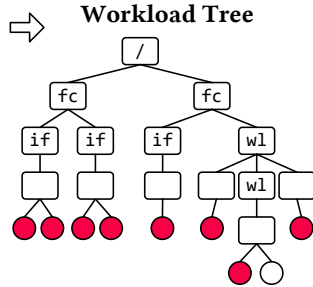
```
X = scale(X, TRUE, TRUE)  
w = l2svm(X, y, TRUE,  
          1e-9, 1e-3, 100)
```

```
write(w, "data/wXy")
```

Built-in Functions:

```
if(shift)  
  X = X - colMeans(X)  
if(scale)  
  X = X / colSds(X)
```

```
if(intercept)  
  X = cbind(X, ones)  
while(conto & i<maxi) {  
  Xd = X %**% s  
  while(conti) {  
    out = 1-y*(Xw+sz*Xd)  
    sz = sz - g/h; # ...  
  }  
  g_new = t(X) %**% (out*y)  
}
```



Workload Awareness



User Script:

```
X = read("data/X")  
y = read("data/y")
```

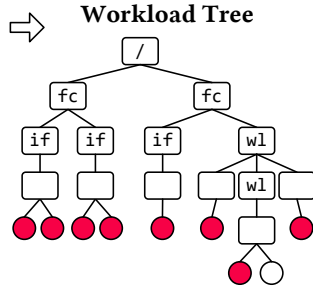
```
X = scale(X, TRUE, TRUE)  
w = l2svm(X, y, TRUE,  
          1e-9, 1e-3, 100)
```

```
write(w, "data/wXy")
```

Built-in Functions:

```
if(shift)  
  X = X - colMeans(X)  
if(scale)  
  X = X / colSds(X)
```

```
if(intercept)  
  X = cbind(X, ones)  
while(conto & i<maxi) {  
  Xd = X %**% s  
  while(conti) {  
    out = 1-y*(Xw+sz*Xd)  
    sz = sz - g/h; # ...  
  }  
  g_new = t(X) %**% (out*y)
```



Cost Summary

0	100	10	10	105	0
---	-----	----	----	-----	---

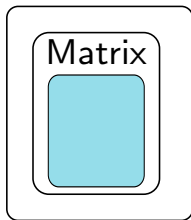
Compression Execution



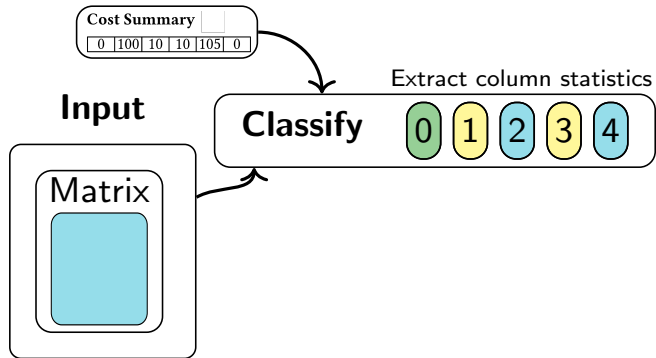
Cost Summary

0	100	10	10	105	0
---	-----	----	----	-----	---

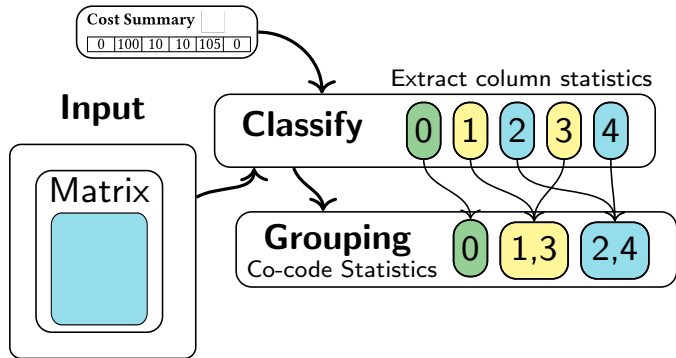
Input



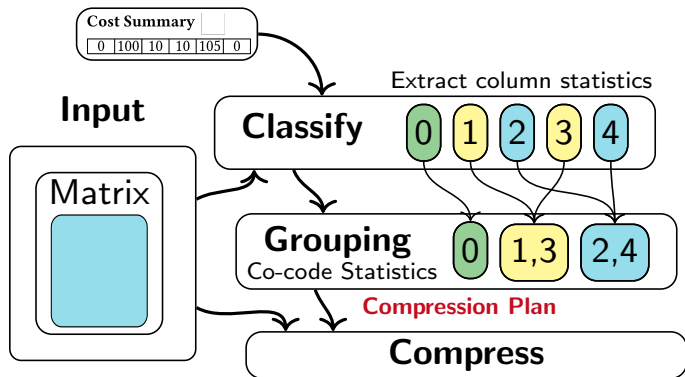
Compression Execution



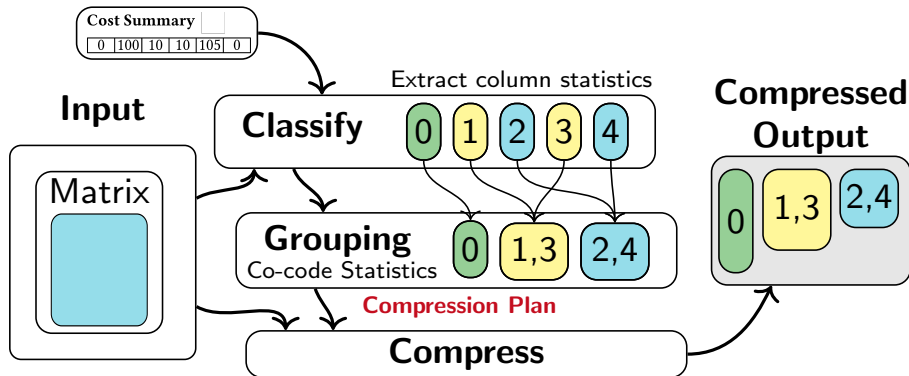
Compression Execution



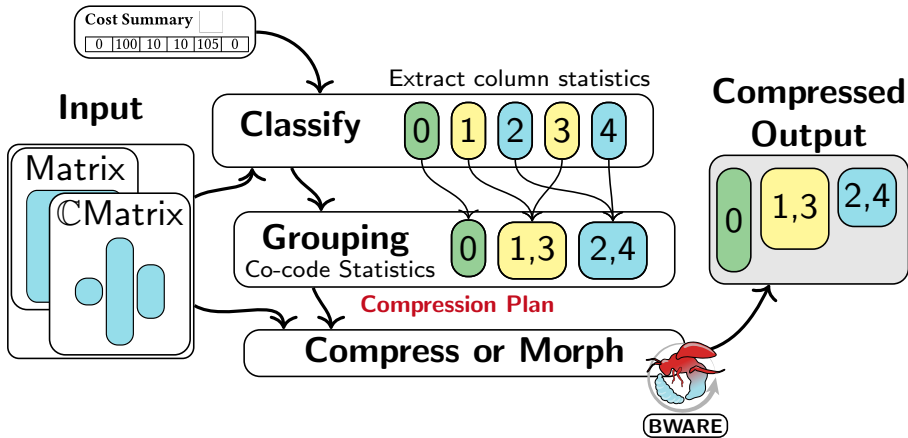
Compression Execution



Compression Execution



Compression Execution

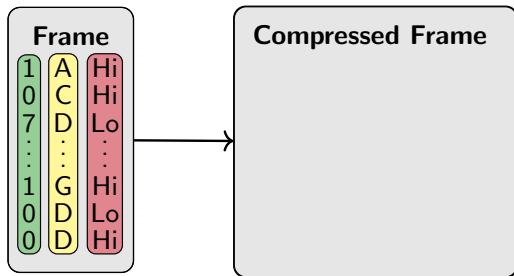


Compressed Frame

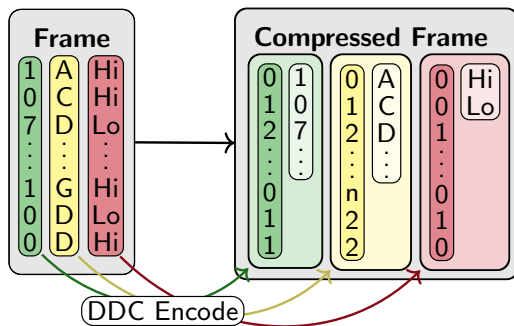


Frame		
1	A	Hi
0	C	Hi
7	D	Lo
:	:	:
:	:	:
1	G	Hi
0	D	Lo
0	D	Hi

Compressed Frame



Compressed Frame

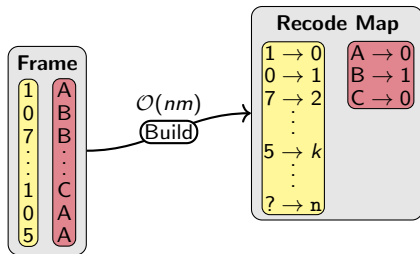


Feature Transformation Example

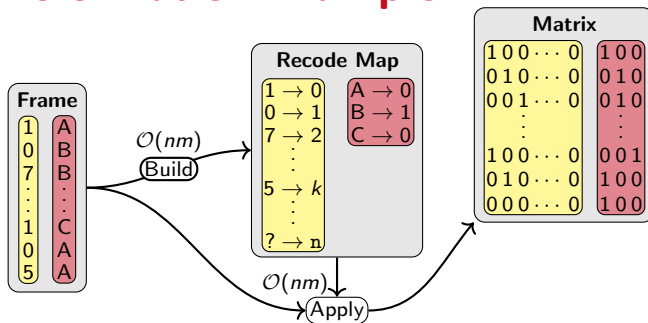


Frame	
1	A
0	B
7	B
...	...
1	C
0	A
5	A

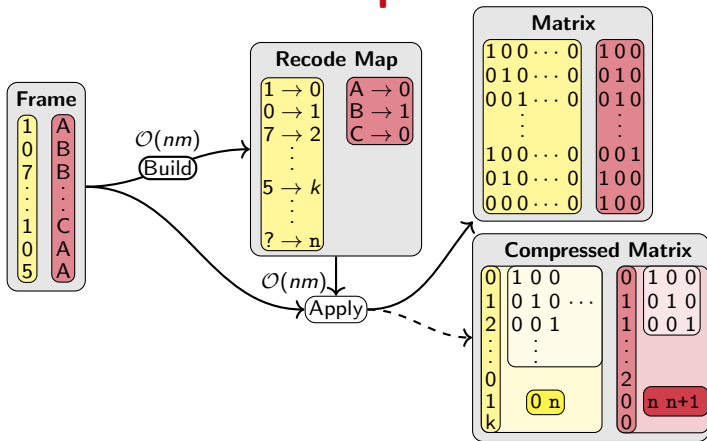
Feature Transformation Example



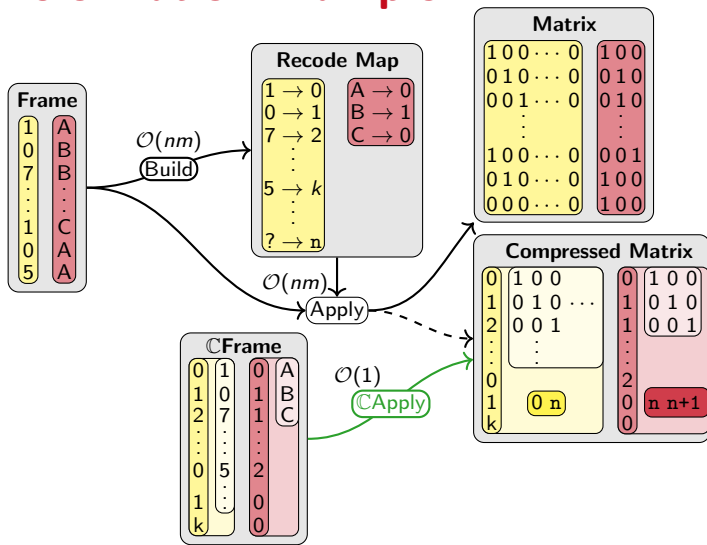
Feature Transformation Example



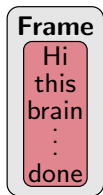
Feature Transformation Example



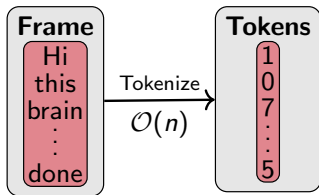
Feature Transformation Example



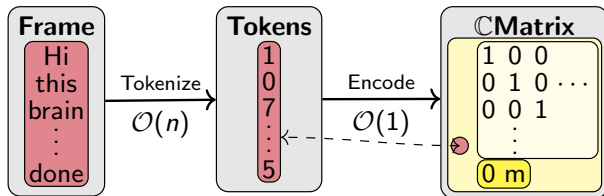
Word Embedding



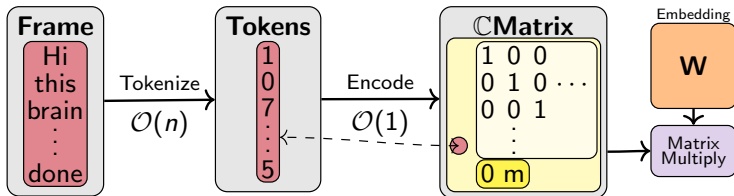
Word Embedding



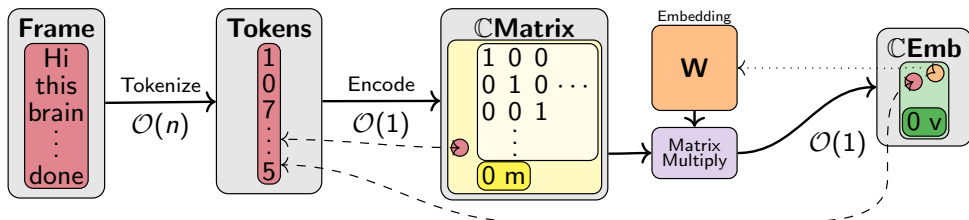
Word Embedding



Word Embedding



Word Embedding



ML Pipeline



Pipeline LM: 8 Transform Encode specifications and 8 Polynomials.

Dataset	Measure	ULA	AWARE	BWARE
KDD98	Execution Time	654s	452s	251s

Pipeline LM: 8 Transform Encode specifications and 8 Polynomials.

Dataset	Measure	ULA	AWARE	BWARE
KDD98	Execution Time	654s	452s	251s
	Compression & Morphing	—	148s	21.9s
	Transform-Encode	74.1s	59.9s	7.95s

Pipeline LM: 8 Transform Encode specifications and 8 Polynomials.

Dataset	Measure	ULA	AWARE	BWARE
KDD98	Execution Time	654s	452s	251s
	Compression & Morphing	—	148s	21.9s
	Transform-Encode	74.1s	59.9s	7.95s
	Instructions	$110 \cdot 10^{12}$	$100 \cdot 10^{12}$	$44 \cdot 10^{12}$
	Instructions per Cycle	0.94	2.59	2.68
	L1 Cache Misses	$7,792 \cdot 10^9$	$1,740 \cdot 10^9$	$786 \cdot 10^9$

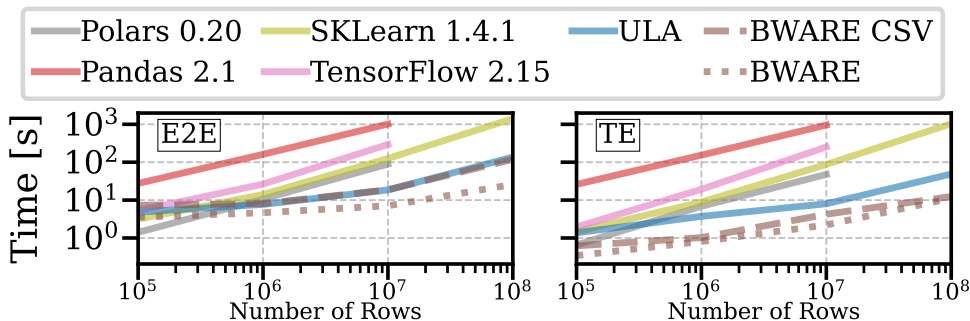
Pipeline LM: 8 Transform Encode specifications and 8 Polynomials.

Dataset	Measure	ULA	AWARE	BWARE
KDD98	Execution Time	654s	452s	251s
	Compression & Morphing	—	148s	21.9s
	Transform-Encode	74.1s	59.9s	7.95s
	Instructions	$110 \cdot 10^{12}$	$100 \cdot 10^{12}$	$44 \cdot 10^{12}$
	Instructions per Cycle	0.94	2.59	2.68
	L1 Cache Misses	$7,792 \cdot 10^9$	$1,740 \cdot 10^9$	$786 \cdot 10^9$
	Energy Consumption	338kJ	185kJ	92kJ

Pipeline LM: 8 Transform Encode specifications and 8 Polynomials.

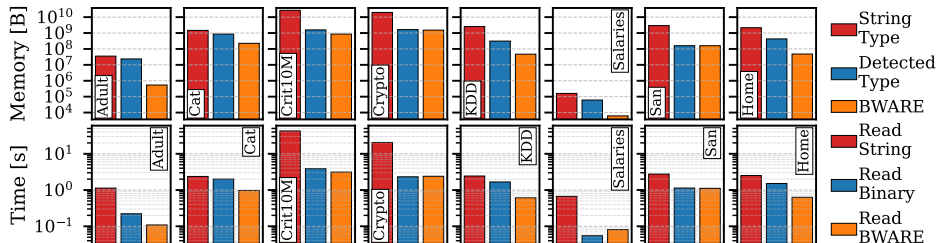
Dataset	Measure	ULA	AWARE	BWARE
KDD98	Execution Time	654s	452s	251s
	Compression & Morphing	—	148s	21.9s
	Transform-Encode	74.1s	59.9s	7.95s
	Instructions	$110 \cdot 10^{12}$	$100 \cdot 10^{12}$	$44 \cdot 10^{12}$
	Instructions per Cycle	0.94	2.59	2.68
	L1 Cache Misses	$7,792 \cdot 10^9$	$1,740 \cdot 10^9$	$786 \cdot 10^9$
	Energy Consumption	338kJ	185kJ	92kJ
Home	Execution Time	431s	266s	160s
Adult	Execution Time	19.9s	22.1s	16.6s
Santander	Execution Time	489s	376s	374s
Cat	Execution Time	467s	170s	63s

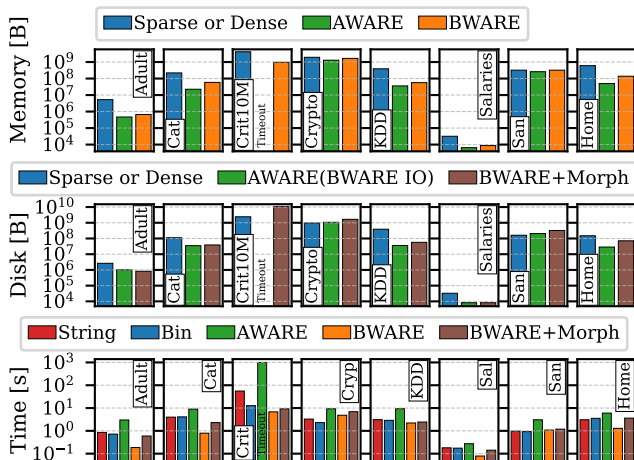
System Comparison



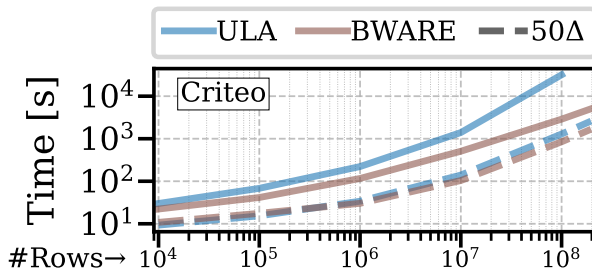
Performance Comparison with Other Systems (Criteo)

Frame Compression

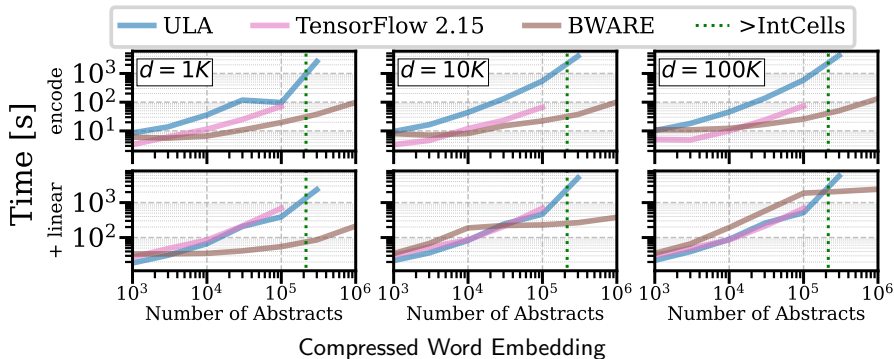




Scaling Performance



Word Embedding



Conclusions



Conclusions



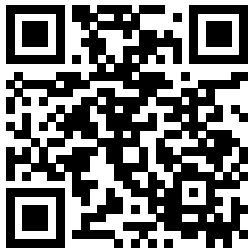
- ❖ Morph compression through feature transformations and engineering.

- ❖ Morph compression through feature transformations and engineering.
 - ❖ Avoid rediscovering compression plans
 - ❖ Get faster end-to-end pipelines
 - ❖ Lower your energy consumption

- ❖ Morph compression through feature transformations and engineering.
 - ❖ Avoid rediscovering compression plans
 - ❖ Get faster end-to-end pipelines
 - ❖ Lower your energy consumption
- ❖ Keep data compressed across the full data-centric ML pipeline.

- ❖ Morph compression through feature transformations and engineering.
 - ❖ Avoid rediscovering compression plans
 - ❖ Get faster end-to-end pipelines
 - ❖ Lower your energy consumption
- ❖ Keep data compressed across the full data-centric ML pipeline.
- ❖ Specialize encodings to the workload via morphing.

Questions?



Contact



Paper



Code



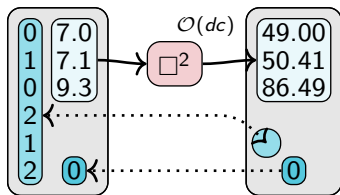
Backup

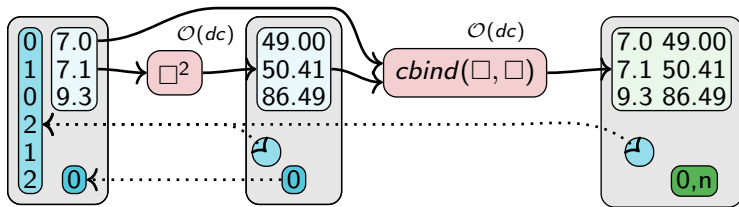
Feature Engineering



0	7.0
1	7.1
0	9.3
2	
1	
2	0

Feature Engineering

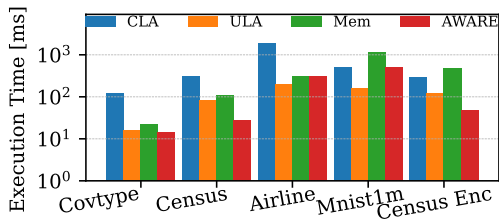




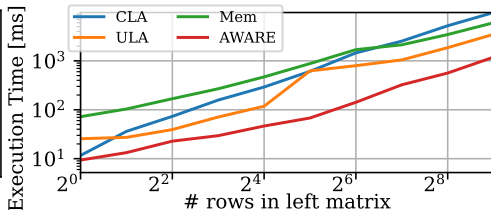
Energy Efficiency



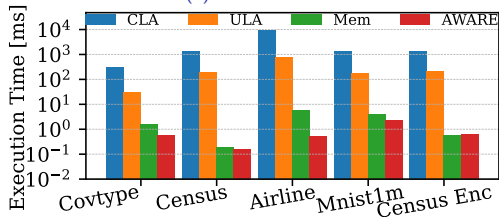
Matrix Multiplication



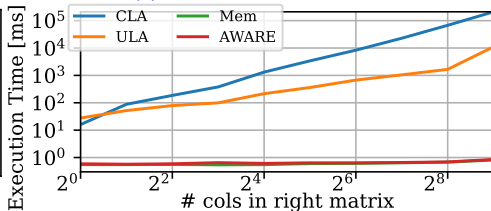
(a) 16 row LMM



(b) LMM Census Enc Scaling

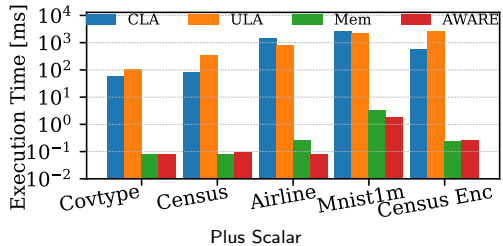


(c) 16 col RMM

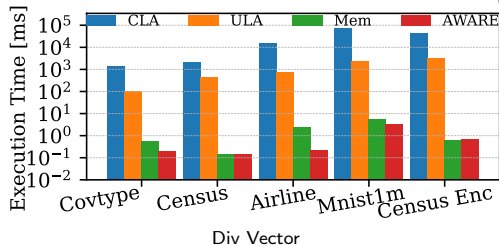
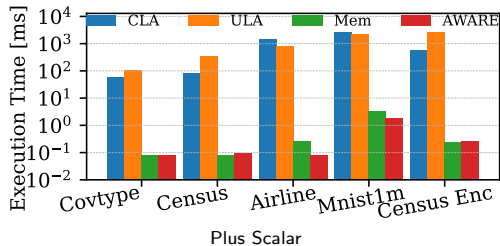


(d) RMM Census Enc Scaling

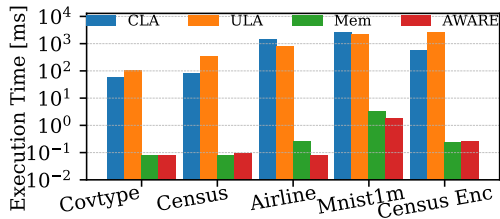
Operation performance



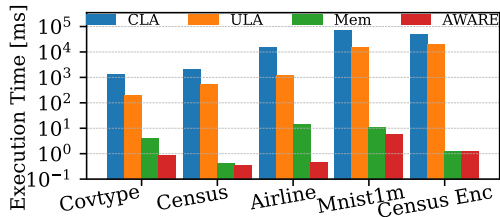
Operation performance



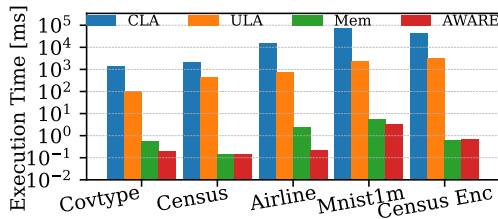
Operation performance



Plus Scalar

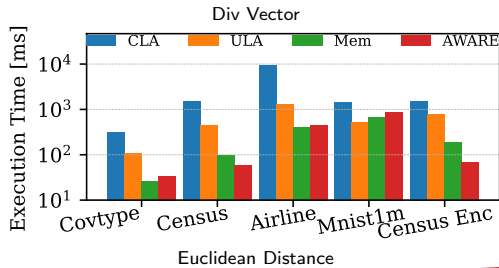
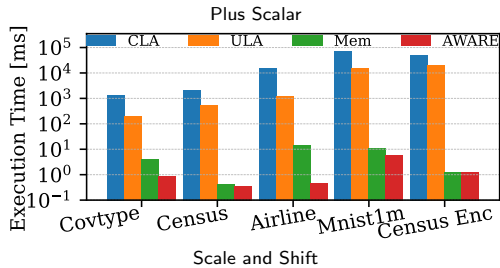
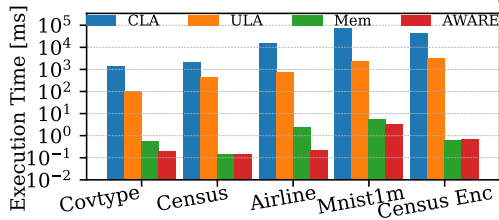
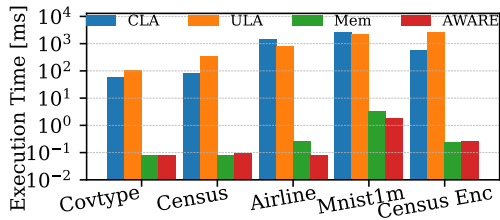


Scale and Shift



Div Vector

Operation performance



BWARE - Transformation Datasets



Name	Rows	Cols	Categorical	Numeric	Task
Adult	32,561	15	9	6	Binary
CatInDat	900,000	24	16	8	Binary
Criteo Day 0	195,841,983	39	25	14	Binary
Crypto	24,236,806	10	1	9	Regression
KDD98	96,367	481	135	334	Regression
Santander	200,000	201	0	201	Binary
HomeCredit	307,511	121	16	105	Binary
Salaries	397	6	3	3	Binary
AMiner V16	5,259,857	1,000	1,000	0	Word Embed

Combine Column Groups



DDC1		DDC2	
0	7.0	0	2.2
1	7.1	1	3.1
0	9.3	0	
2		1	
1		1	
2	4	1	8

Combine Column Groups



DDC1		DDC2	
0	7.0	0	2.2
1	7.1	1	3.1
0	9.3	0	
2		0	
1		1	
2		1	
4		8	

Naive			
0	7.0	2.2	4,8
3	7.0	3.1	
0	7.1	2.2	
5	7.1	3.1	
3	9.3	2.2	
5	9.3	3.1	

Combine Column Groups

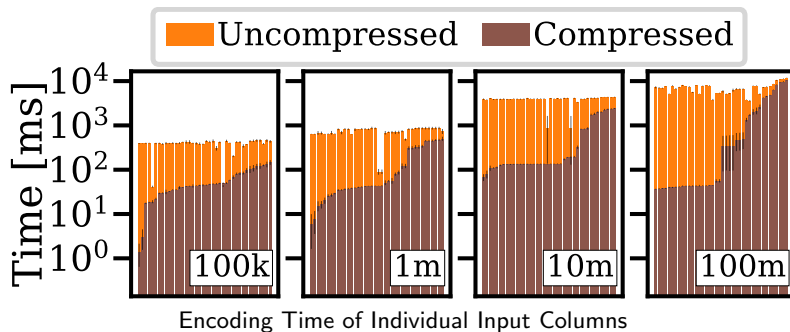


DDC1		DDC2	
0	7.0	0	2.2
1	7.1	1	3.1
0	9.3	0	
2		0	
1		1	
2		1	
4		8	

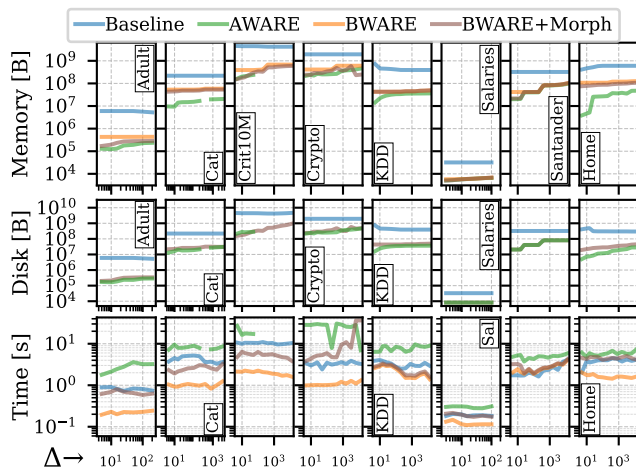
Naive			
0	7.0	2.2	4,8
3	7.0	3.1	
0	7.1	2.2	
5	7.1	3.1	
3	9.3	2.2	
5	9.3	3.1	

Optimal		
0	7.0	2.2
1	7.1	3.1
0	9.3	3.1
2		4,8
1		
2		

Transformation Breakdown



Lossy Performance



Extended Pipeline



8 transformations, 8 Polynomials, MICE, and winsorize

Dataset	Measure	ULA	BWARE
KDD98	Execution Time	919.6s	255.3s

Extended Pipeline



8 transformations, 8 Polynomials, MICE, and winsorize

Dataset	Measure	ULA	BWARE
KDD98	Execution Time	919.6s	255.3s
	Compression & Morphing	—	24.9s
	Transform-Encode	37.7s	5.4s

Extended Pipeline



8 transformations, 8 Polynomials, MICE, and winsorize

Dataset	Measure	ULA	BWARE
KDD98	Execution Time	919.6s	255.3s
	Compression & Morphing	—	24.9s
	Transform-Encode	37.7s	5.4s
	Instructions	$103 \cdot 10^{12}$	$45 \cdot 10^{12}$
	Instructions per Cycle	0.82	2.68
	L1 Cache Misses	$8,276 \cdot 10^9$	$830 \cdot 10^9$

Polynomial Performance

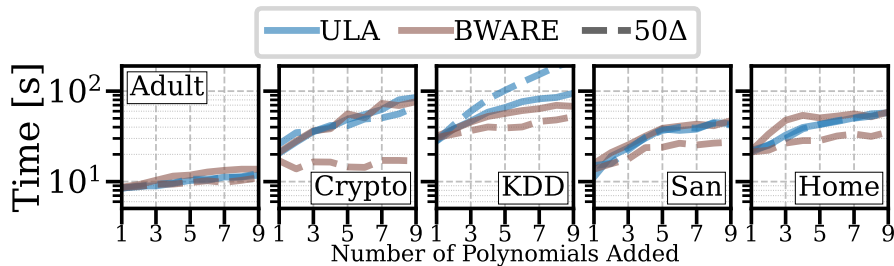


Table Transform-Encode Column Asymptotic Size.

	F-M	F-CM	CF-CM
Recode & Pass Bin & Hash	$8 \mathbf{X} $ $8 \mathbf{X} $	$\#B \mathbf{X} + 8d_i$ $\#B \mathbf{X} + 8\Delta$	<i>constant</i> $\#B \mathbf{X} + 8\Delta$
With One-Hot / Dummy-Coding			
Recode & Pass Bin & Hash	$12 \mathbf{X} $ $12 \mathbf{X} $	$\#B \mathbf{X} $ $\#B \mathbf{X} $	<i>constant</i> $\#B \mathbf{X} $
Word Embed	$(8v + 12) \mathbf{X} $	$\#B \mathbf{X} $	constant

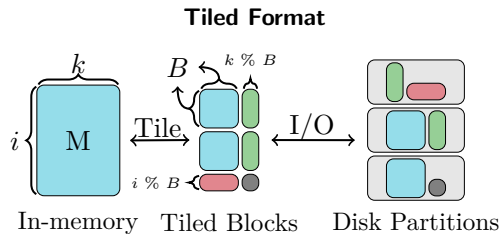
On Disk — Tiled Format



Tiled Format

Compressed Tiled Format

On Disk — Tiled Format



Compressed Tiled Format

On Disk — Tiled Format

