Mitigating Forgetting in LLM via Low-Perplexity Token Learning

Chao-Chung Wu [1], Zhi Rui Tam [1, 2], Chieh-Yen Lin [1],
Yun-Nung Chen [2], Shao-Hua Sun [1, 2], Hung-yi Lee [2]

[1]Appier AI Research

[2]National Taiwan University

Appier

Background

- High quality fine-tuning data **distilled** by LLM brings significant improvements of **higher target task** performance and **less non-target task forgetting**[1].

- It is not explored how generated data is favorable of training even though the data brings **different context**, generally fits **different scales of models** and is not intentionally trained for **recovering non-target tasks**...

[1] Zelikman et. al Star: Bootstrapping reasoning with reasoning

TL;DR

In this paper, we

- bridge **perplexity** of tokens as a criteria of training data with the phenomena of **catastrophic forgetting after training**.
- proposed **STM**, a practical, low-cost alternative to expensive synthetic generated data methods for maintaining competitive performance after training.
- found **STM's generalization** across models scales & families, training strategies and different domains of tasks.

Typical LLM generated data

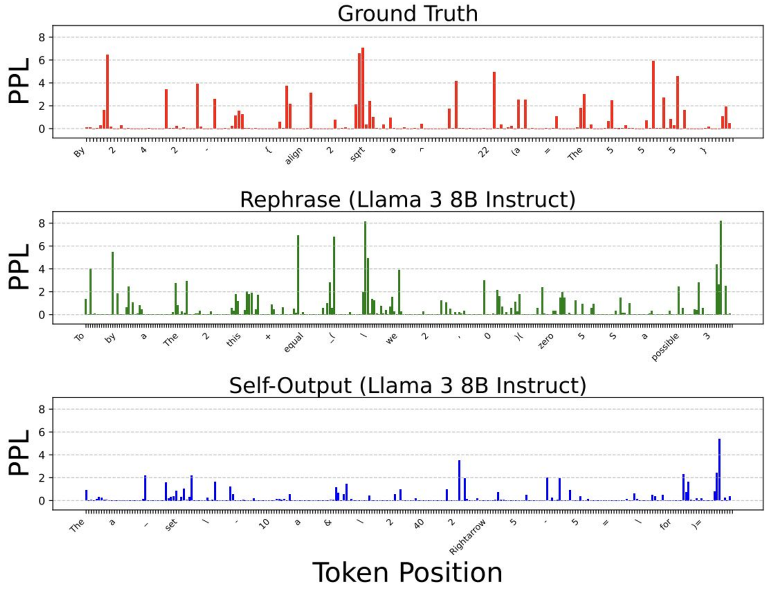
- **Self-Output [Ren et al. and Trummer]**
  - High quality synthetic label output of LLM, which is verified by ground truth answers.

- **Rephrase [Yang et al.]**
  - LLM output by giving instruction and ground truth, which is also verified by ground truth sanity check.

- Generated data v.s. Human annotation (Ground Truth)?
  - need **verification on answers** (reasoning path is harder to check)
  - **hallucination** from LLM generation

Why LLM generated data might improve performance?

- Distillation brings **low-perplexity** intrinsically. e,g., math arithmetic reasoning, coding tasks, knowledge QA tasks.
- Low perplexity implies **fewer model parameter changing** of training → fewer non related tasks are affected.



Data	Method	Avg. PPL
MBPP	Ground Truth	4.83 (7.04)
	Rephrase	1.69 (0.16)
	Self-Output	1.16 (0.01)
MATH	Ground Truth	2.45 (0.81)
	Rephrase	2.11 (9.28)
	Self-Output	1.34 (0.03)

A critical criteria of data to affect forgetting: Perplexity

- Perplexity difference in Human annotated data and LLM generated data
- problem: How to scale perplexity without changing context of training data?

Question: Given a mathematics problem, determine the answer.\nProblem : Let the reciprocals of the roots of $5x^2 + 3x + 4$ be α and β. Evaluate $\alpha + \beta$.\nAnswer:$

Ground Truth: Denote the roots of $5x^2 + 3x + 4$ by a and b. We have that $\alpha = \frac{1}{a}$ and $\beta = \frac{1}{b}$. So, $\alpha + \beta = \frac{1}{a} + \frac{1}{b} = \frac{a+b}{ab}$. Now, we know that $a+b = \frac{-3}{5}$ and $ab = \frac{4}{5}$ by the relationship between sum/products of roots and the coefficients of a polynomial. Hence $\alpha + \beta = \frac{a+b}{ab} = \boxed{-\frac{3}{4}}$

Self-Output (Llama 3 8B Instruct): Let r_1 and r_2 be the roots of $5x^2 + 3x + 4$. Then, by Vieta's formulas, we have $r_1 + r_2 = -\frac{3}{5}$ and $r_1 r_2 = \frac{4}{5}$. Let $\alpha = \frac{1}{r_1}$ and $\beta = \frac{1}{r_2}$. Then, we have $\alpha + \beta = \frac{1}{r_1} + \frac{1}{r_2} = \frac{r_2 + r_1}{r_1 r_2} = \frac{-\frac{3}{5}}{\frac{4}{5}} = \boxed{-\frac{3}{4}}$

STM: Selective Token Masking

- Select high perplexity tokens with a **fixed criteria, say 2.5**
- **Masking high PPL** tokens in fine-tuning stage to block back propagation.

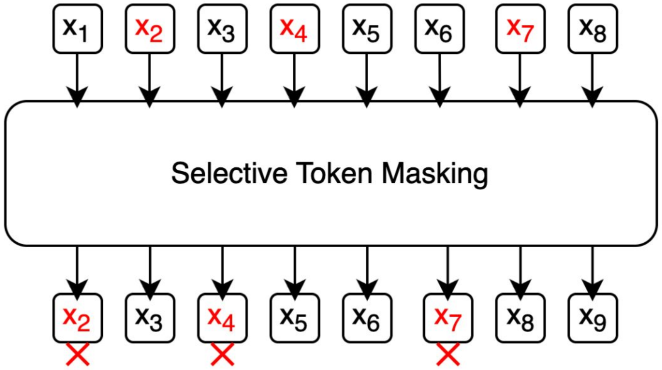
User: Let the reciprocals of the roots of $5x^2 + 3x + 4$ be α and β. Evaluate α + β

Assistant: Denote the roots of $5x^2 + 3x + 4$ by a and b. We have that α = 1/a and β =1/b. So, α + β = 1/a + 1/b = (a+b)/(ab). Now, we know that a+b = -⅗ and ab = ⅘ [redacted] between [redacted] of roots and [redacted] coefficients of a [redacted]. [redacted]ence α + β = ([redacted]+b)/ab = [redacted]

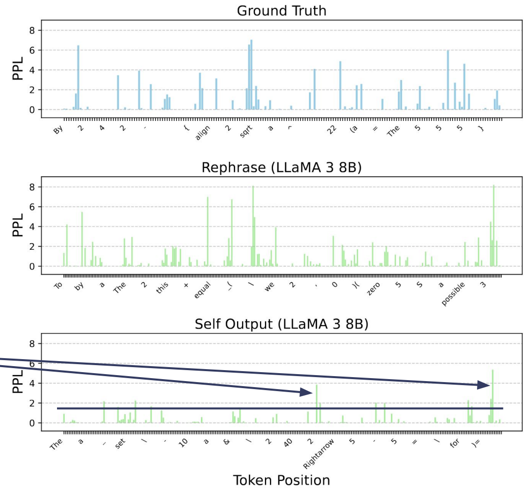
STM: Selective Token Masking

- A simple yet efficient way to filter high ppl tokens from GT data.
  - **no need for pretraining** an LLM/reference model from high quality data or low loss data.

Den ote the roots of $5x^2$ + 3 x + 4 as a and b.We have that \apha = \frac{1} {a}, \beta = \frac{1} {b}



skip these tokens



$$\mathcal{L}_{\mathrm{STM}}(\theta) = \frac{\sum_{t=1}^{N} \mathbb{1}\{-\log p_\theta(w_t \mid w_{<t}) \le \log \tau\} \left(-\log p_\theta(w_t \mid w_{<t})\right)}{\max\left(1, \sum_{t=1}^{N} \mathbb{1}\{-\log p_\theta(w_t \mid w_{<t}) \le \log \tau\}\right)}$$

Appier

Experiment: Is STM general to different model & scales?

- comparable results of STM to Self-Output performance in terms of
  - target improvement (TI)
  - changes on non-target degradation (BWT)
- From **2B~9B across** Llama 3, Gemma 2, Mistral, OLMo 2 model families
  - **MBPP** (coding task) and **MATH** (arithmetic reasoning) as training tasks.
  - **testing data** (MBPP or MATH), **GSM8k**, **BIRD**, **IFEval**, **safety** as testing tasks for TI and BWT calculation

$$\mathbf{TI} = (a_{target}^{(train)} - a_{target}^{(original)})/a_{target}^{(original)}.$$

$$\mathbf{BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1}(a_i^{(train)} - a_i^{(original)})/a_i^{(original)}.$$

Experiment: Is STM general to different model & scales?

- From 2B~9B across **Llama 3, Gemma 2,** Mistral, OLMo 2 model families

Model	Target task	Method	BWT(%)	TI(%)	Cost (GPU hours)
Gemma 2 IT 2B	MBPP	Baseline Fine-tuning	-38.19	-21.76	0
		Self-Output	-8.10	5.70	12 Hours
		Rephrase	-3.23	-4.69	30 Minutes
		STM$_{\tau=2.5}$ (Ours)	**0.42**	0.00	5 Minutes
	MATH	Baseline Fine-tuning	-36.68	-22.78	0
		Self-Output	**-1.73**	9.06	$\geq$ 2 Days
		Rephrase	-14.06	-28.83	39 Minutes
		STM$_{\tau=2.5}$ (Ours)	**-2.93**	7.83	8 Minutes
Llama 3 8B Instruct	MBPP	Baseline Fine-tuning	-34.71	-2.23	0
		Self-Output	**3.09**	1.55	16.8 Hours
		Rephrase	-5.32	-9.58	36.8 Minutes
		STM$_{\tau=2.5}$ (Ours)	**-0.16**	3.20	4.5 Minutes
	MATH	Baseline Fine-tuning	-14.12	-17.83	0
		Self-Output	**0.31**	9.55	$\geq$2 Days
		Rephrase	-1.09	4.78	29.3 Minutes
		STM$_{\tau=2.5}$ (Ours)	**-0.30**	6.37	7 Minutes

Experiment: Is STM general to different model & scales?

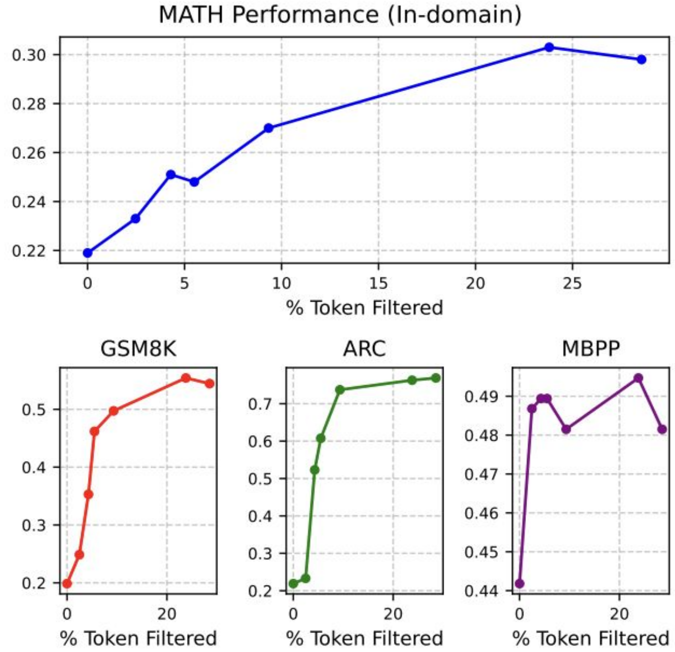
- From 2B~9B across Llama 3, Gemma 2, Mistral, **OLMo 2** model families

New Model	TI (%)	BWT (%)
OLMo 2 7B Instruct		
Baseline Fine-tuning	-11.64	-9.05
$STM_{\tau=2.5}$ (25.83%)	**-6.12**	**-0.50**
Gemma 2 IT 9B		
Baseline Fine-tuning	7.14	**4.67**
$STM_{\tau=2.5}$ (20.84%)	**13.49**	2.58

Optimal choice of PPL threshold and PPL calculation

- filter out about **20~24% of high ppl tokens**, stm gains it in-domain and out-of-domain task performance optimally.
- calculation of ppl by the **same model is better than a larger one.**

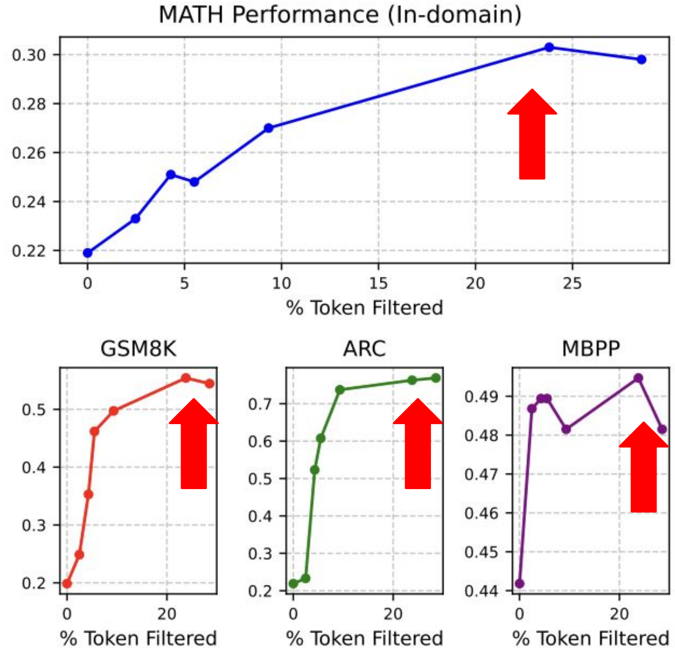
Configuration	BWT(%)	TI(%)
$\text{STM}_{\tau=2.5,high}$	**0.4**	**0.0**
$\text{STM}_{\tau=2.5,random}$	-8.6	-15.6
$\text{STM}_{\tau=2.5,low}$	-7.9	-18.7
Baseline Fine-tuning	-38.2	-25.2
$\text{STM}_{\tau=1000}$ (6.26%)	-2.9	-11.4
$\text{STM}_{\tau=25}$ (12.34%)	-2.5	-8.8
$\text{STM}_{\tau=10}$ (15.1%)	-0.7	-10.4
$\text{STM}_{\tau=2.5}$ (23.8%)	**0.4**	**0.0**
$\text{STM}_{\tau=1.5}$ (26.1%)	-0.3	-0.5
$\text{STM}_{9B\tau=2.5}$ (23.8%)	-3.8	-7.3



Optimal choice of PPL threshold and PPL calculation

- filter out about **20~24% of high ppl tokens**, stm gains it in-domain and out-of-domain task performance optimally.
- calculation of ppl by the **same model is better than a larger one.**

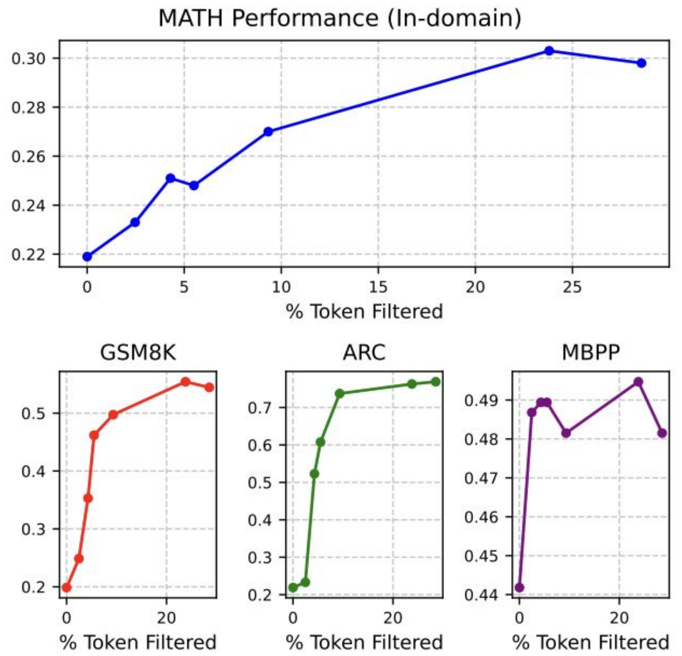
Configuration	BWT(%)	TI(%)
$\text{STM}_{\tau=2.5,high}$	**0.4**	**0.0**
$\text{STM}_{\tau=2.5,random}$	-8.6	-15.6
$\text{STM}_{\tau=2.5,low}$	-7.9	-18.7
Baseline Fine-tuning	-38.2	-25.2
$\text{STM}_{\tau=1000}$ (6.26%)	-2.9	-11.4
$\text{STM}_{\tau=25}$ (12.34%)	-2.5	-8.8
$\text{STM}_{\tau=10}$ (15.1%)	-0.7	-10.4
$\text{STM}_{\tau=2.5}$ (23.8%)	**0.4**	**0.0**
$\text{STM}_{\tau=1.5}$ (26.1%)	-0.3	-0.5
$\text{STM}_{9B\tau=2.5}$ (23.8%)	-3.8	-7.3



Optimal choice of PPL threshold and PPL calculation

- filter out about **20~24% of high ppl tokens**, stm gains it in-domain and out-of-domain task performance optimally.
- calculation of ppl by the **same model is better than a larger one.**

Configuration	BWT(%)	TI(%)
$\text{STM}_{\tau=2.5,high}$	**0.4**	**0.0**
$\text{STM}_{\tau=2.5,random}$	-8.6	-15.6
$\text{STM}_{\tau=2.5,low}$	-7.9	-18.7
Baseline Fine-tuning	-38.2	-25.2
$\text{STM}_{\tau=1000}$ (6.26%)	-2.9	-11.4
$\text{STM}_{\tau=25}$ (12.34%)	-2.5	-8.8
$\text{STM}_{\tau=10}$ (15.1%)	-0.7	-10.4
$\text{STM}_{\tau=2.5}$ (23.8%)	**0.4**	**0.0**
$\text{STM}_{\tau=1.5}$ (26.1%)	-0.3	-0.5
$\text{STM}_{9B\tau=2.5}$ (23.8%)	-3.8	-7.3



Stable performance across learning rate

- Lower sensitivity to learning rate of STM than that of baseline fine-tuning.

Llama 3 8B-IT	lr	BWT(%)	TI(%)
BASELINE	1E-4	-	-10.6
BASELINE	2E-5	-34.7	-2.23
BASELINE	5E-6	-	0.9
BASELINE	1E-6	-	-4.47
BASELINE	5E-7	-	-1.3
BASELINE	1E-7	-1.6	1.33
STM	1E-4	1.8	-3.58
STM	2E-5	0.2	3.2
STM	5E-6	-0.1	2.68
STM	1E-6	0.53	3.12
STM	5E-7	1.23	3.12
STM	1E-7	1.39	2.23

TI varies

stable & positive TI/BWT

Gemma 2 IT 2B	lr	BWT(%)	TI(%)
BASELINE	2E-5	-38.2	-15.5
BASELINE	5E-6	-	-4.0
BASELINE	1E-6	-	-17.6
BASELINE	1E-7	-4.7	-0.53
STM	2E-5	-0.3	-0.5
STM	5E-6	-1.1	-3.0
STM	1E-6	-0.35	-1.5
STM	1E-7	0.51	0.7

Generalization of STM across training strategies

- STM **enhance** all the training techniques (full weight and parameter efficient training):
  - Full weight fine-tuning
  - Lora fine-tuning
  - Dora fine-tuning

Configuration	BWT (%)	TI(%)
FWFT	-31.87	-27.98
FWFT + $STM_{\tau=2.5}$	**-0.13**	**-8.81**
LoRA	-21.76	-38.19
LoRA + $STM_{\tau=2.5}$	**0.42**	**0.0**
DoRA	-8.54	-15.2
DoRA + $STM_{\tau=2.5}$	**-0.01**	**-0.04**

Appier

Training with mask == low diversity generation?

- self-bleu scores on 100 MATH testing set.
  - generate 5 samples for each testing instance.
- Both baseline fine-tuning and SFT with STM have similar diversity of generation.
  - **SFT decreases diversity, but diversity doesn't deteriorate with STM..**
- STM **enhances performance** at the same time.

Llama 3 8B-IT	lr	self-bleu	TI(%)	BWT (%)
Original	-	20.47±13	-	-
Baseline	1E-7	40.29±19	26.3	-1.6
$STM_{\tau=2.5}$	1E-7	40.77±17	30.2	1.39

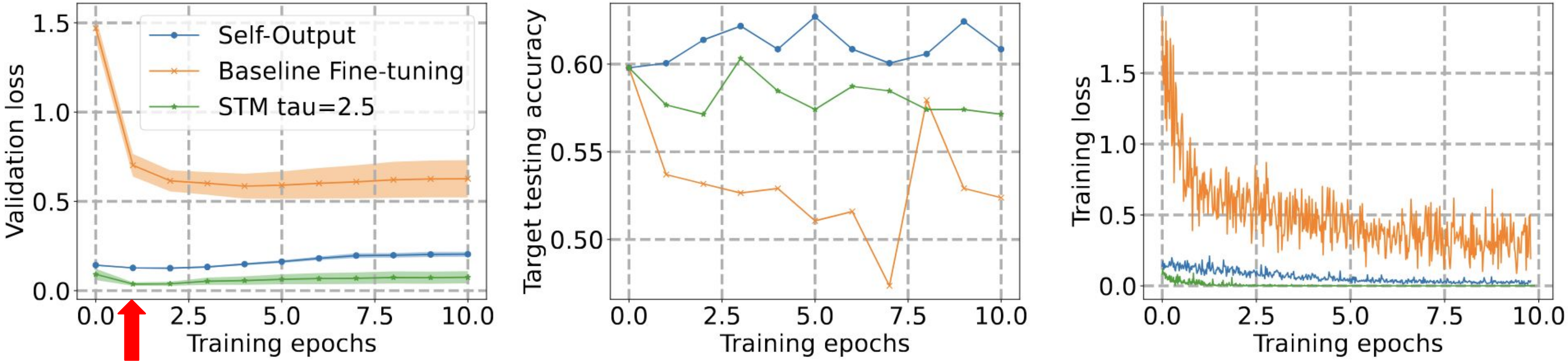
Is STM comparable to regularization?

- Training for fewer parameter changes like **regularization** does not lead to better performance than STM.
  - **weight decay** (sweeping on decay value and dropout)
  - **KL divergence** for L2 regularization.

Regularization & Hyperparameter	L2 norm of ΔW	BWT (%)	TI (%)
WEIGHT DECAY 0 + DROPOUT 0.05	0.7539	-9.24	-9.82
WEIGHT DECAY 0.2 + DROPOUT 0.3	0.7109	-3.50	-8.03
WEIGHT DECAY 0.5 + DROPOUT 0.3	0.5351	-11.15	2.86
KL $coef$ =1E-5	0	-0.24	2.24
STM	0.5500 best	**1.90**	**3.12**

Analysis:Fewer parameter changes in STM training

- Faster **convergence**, fewer **parameter (weight) changes**
- Fewer parameter changes lead to less forgetting intrinsically.



Models tuned on MBPP	Self-Output	Rephrase	Ground Truth	STM + Ground Truth
Llama 3 8B Instruct	6.53	7.31	17.75	0.55
Gemma 2 IT 2B	4.03	5.78	5.69	0.45

Conclusion

1. Selective token masking, STM, bridges the low perplexity of data and one of the reason of LLM catastrophic forgetting after fine-tuning.
2. STM provides a simple and cost-effective alternative to synthetic data training
3. STM shows the generalization across different model scales, model families, training parameters and training strategies.

Thank You

johnson.wu@appier.com, ray.tam@appier.com, coha.lin@appier.com