TimeFlow: an Implicit Neural Representation Approach for Continuous Time Series Modeling

Etienne Le Naour [1,2], Louis Serrano [1]*, Léon Migus [1]*, Yuan Yin [1]*

Ghislain Agoua [2], Nicolas Baskiotis [1], Patrick Gallinari [1,3], Vincent Guigue [4]

1. Sorbonne Université | 2. EDF R&D | 3. Criteo AI Lab | 4. AgroParisTech

Context

1. **Real data challenges:** In real-life applications, time series suffer from irregular sampling and missing data.
2. **Limitations of existing models:** Current models, such as transformers or convolution networks, have difficulty handling irregular data, resulting in a loss of performance and applicability.
3. **Proposed approach:** We propose TimeFlow, a method using conditional implicit neural representations coupled with meta-learning to continuously model time series and manage imputation and forecasting with irregular observations.
4. **Key contributions:** TimeFlow outperforms existing models for imputation and forecasting, offering a flexible solution for diverse real-life scenarios.

TimeFlow

- TimeFlow is a unified framework for imputation and forecasting..
- TimeFlow models time-continuous functions based on implicit neural networks (INR) to represent series.
- INRs are conditioned by modulations to adapt to different series/contexts.
- Time series encoding is achieved through meta-learning optimization.
- Inference is achieved by optimizing codes according to the observed context.

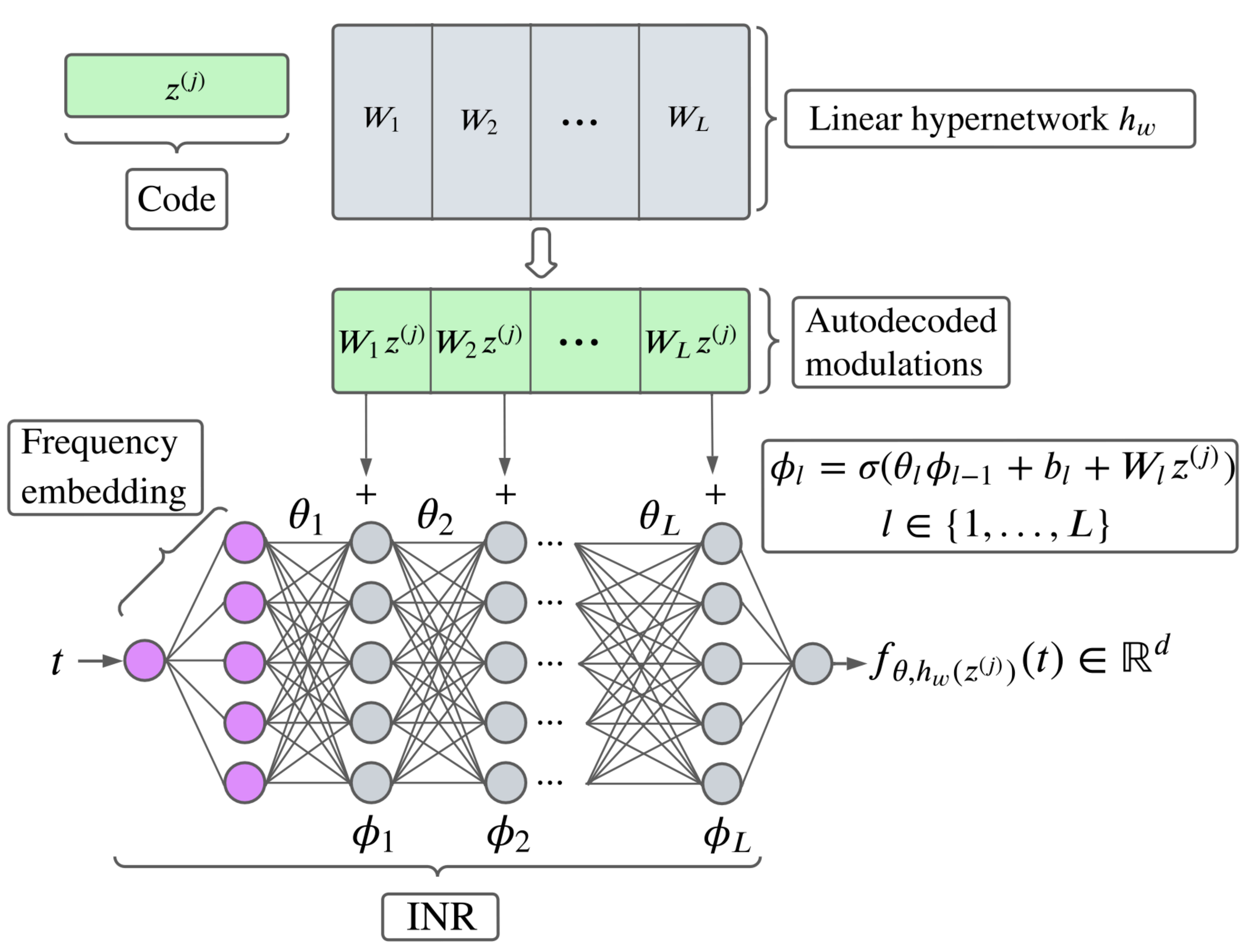


Figure 1: Overview of TimeFlow architecture. Forward pass to approximate the time series $x^{(j)}$. σ stands for the ReLU activation function.

Optimization et meta-learning

TimeFlow training is performed by optimizing shared INR parameters and hypernetwork parameters, as well as series-specific codes, using a meta-learning optimization approach. Individual contexts are used to adjust codes in an inner loop, while shared parameters are updated in an outer loop.

Algorithm 1: TimeFlow Training

while *no convergence* **do**
  Sample batch $\mathcal{B}$ of data $(x^{(j)})_{j \in \mathcal{B}}$;
  Set codes to zero $z^{(j)} \leftarrow 0, \forall j \in \mathcal{B}$;
  // inner loop for encoding:
  for $j \in \mathcal{B}$ *and step* $\in \{1, ..., K\}$ **do**
    $z^{(j)} \leftarrow z^{(j)} - \alpha \nabla_{z^{(j)}} \mathcal{L}_{\mathcal{T}_{in}^{(j)}}(f_{\theta, h_w(z^{(j)})}, x^{(j)})$;
  // outer loop step:
  $[\theta, w] \leftarrow [\theta, w] - \eta \nabla_{[\theta, w]} \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} [\mathcal{L}_{\mathcal{T}_{in}^{(j)}}(f_{\theta, h_w(z^{(j)})}, x^{(j)}) + \lambda \mathcal{L}_{\mathcal{T}_{out}^{(j)}}(f_{\theta, h_w(z^{(j)})}, x^{(j)})]$;

TimeFlow inference adjusts series-specific codes according to observed data, while keeping the shared model parameters fixed. This enables efficient prediction of time series values over new periods.

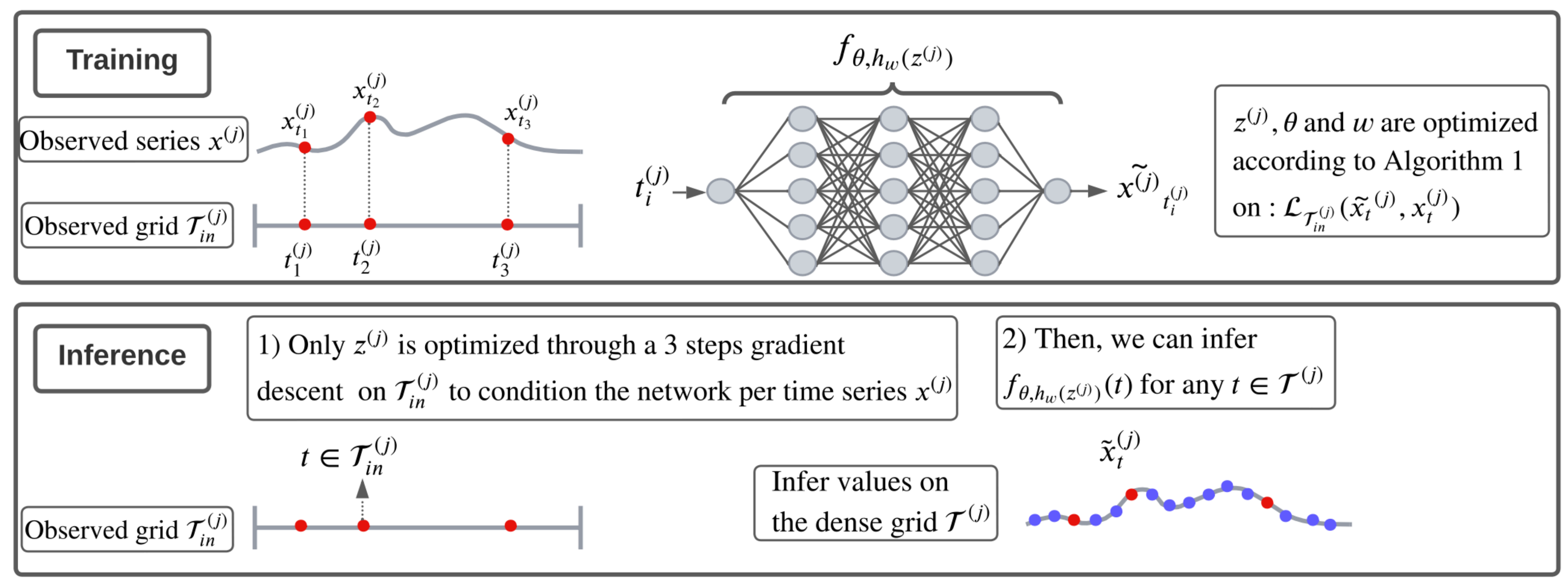
Algorithm 2: TimeFlow Inference with trained θ, w

For the j-th series $(x^{(j)})$, set code to zero $z^{*(j)} \leftarrow 0$;
for *step* $\in \{1, ..., K\}$ **do**
  $z^{*(j)} \leftarrow z^{*(j)} - \alpha \nabla_{z^{*(j)}} \mathcal{L}_{\mathcal{T}_{in}^{*(j)}}(f_{\theta, h_w(z^{*(j)})}, x_t)$
Query $f_{\theta, h_w(z^{*(j)})}(t)$ for any $t \in \mathcal{T}^{*(j)}$

* equal contribution

Imputation experiment

Setting

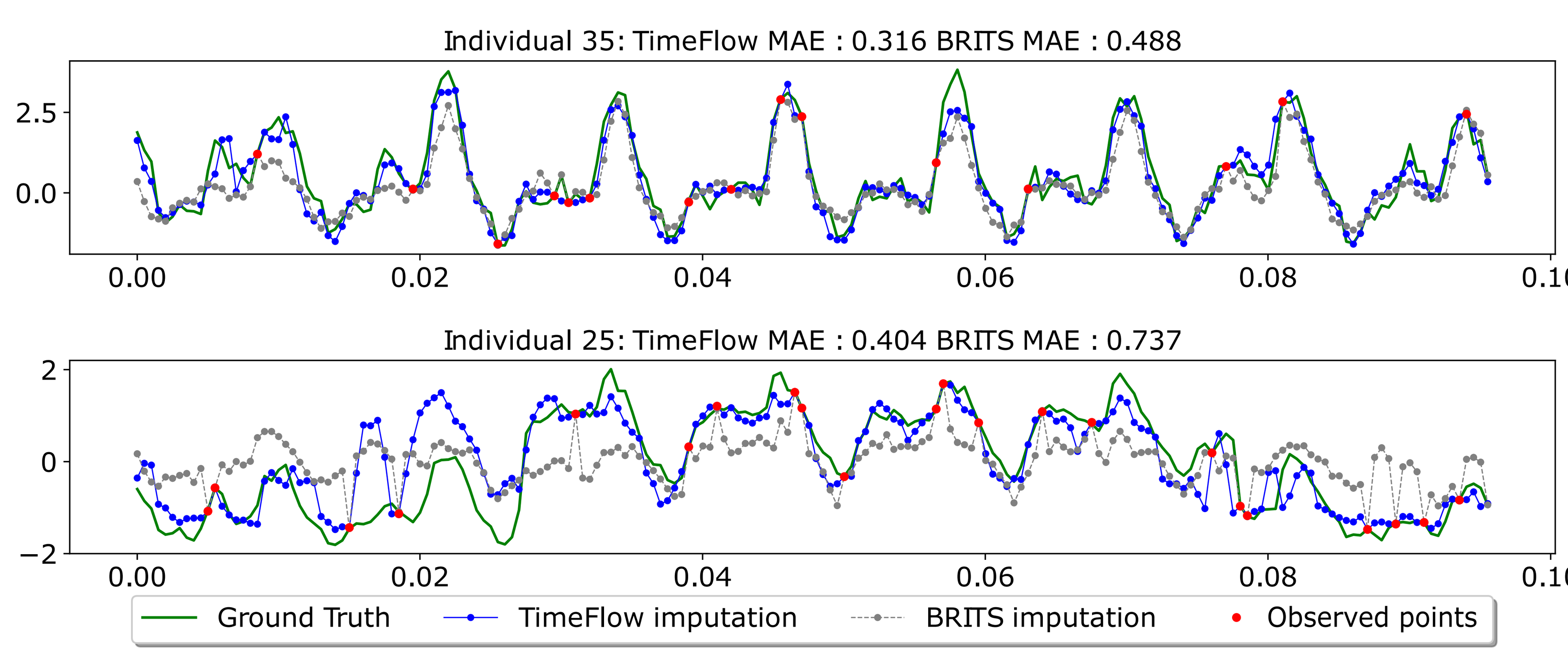


Quantitative results

Table 1: Mean MAE imputation results on the missing grid only. Each time series is divided into 5 time windows onto which imputation is performed, and the performances are averaged over the 5 windows. In the table, τ stands for the subsampling rate, i.e. the proportion of observed points considered for each time window. Bold results are best, underlined results are second best. TimeFlow improvement represents the overall percentage improvement achieved by TimeFlow compared to the specific method being considered.

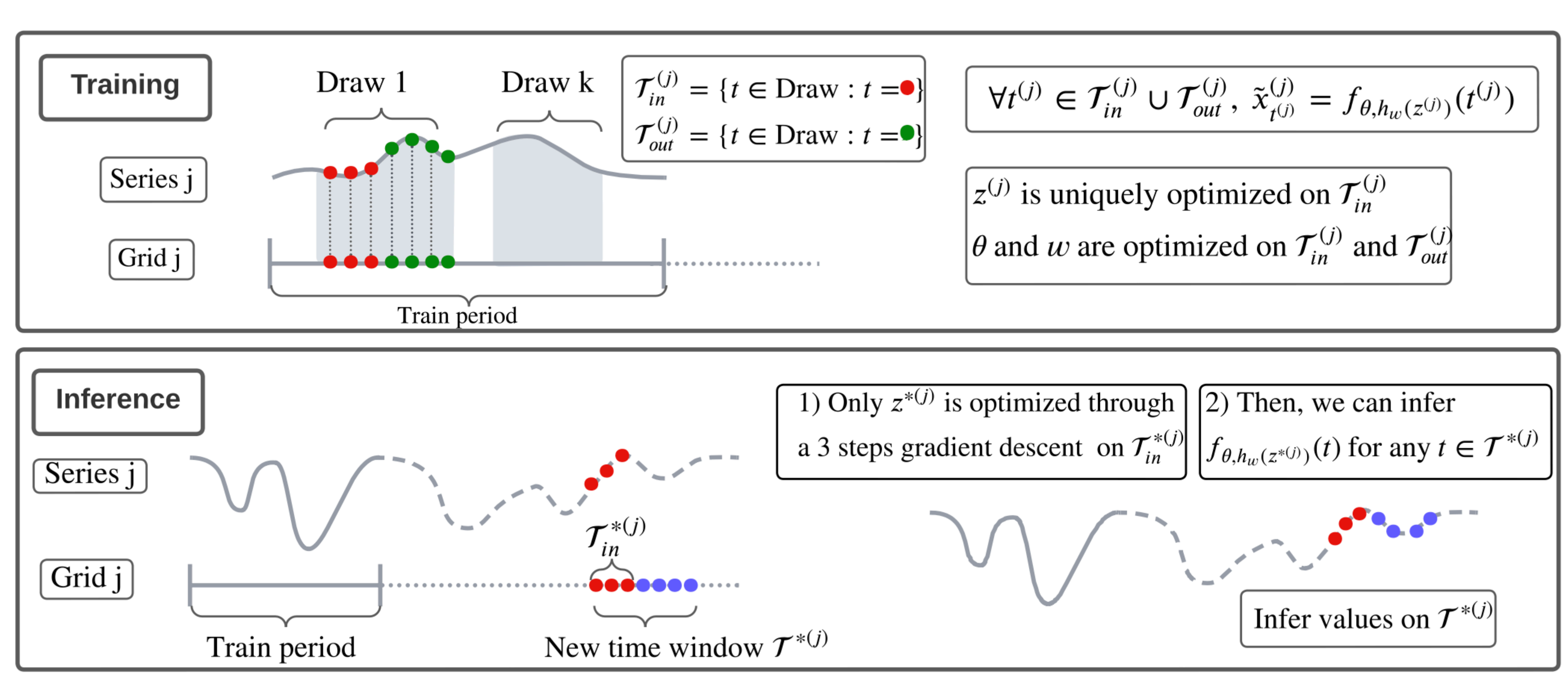
		Continuous methods				Discrete methods			
	τ	TimeFlow	DeepTime	mTAN	Neural Process	CSDI	SAITS	BRITS	TIDER
Electricity	0.05	**0.324 ± 0.013**	0.379 ± 0.037	0.575 ± 0.039	0.357 ± 0.015	0.462 ± 0.021	0.384 ± 0.019	<u>0.329 ± 0.015</u>	0.427 ± 0.010
	0.10	**0.250 ± 0.010**	0.333 ± 0.034	0.412 ± 0.047	0.417 ± 0.057	0.398 ± 0.072	0.308 ± 0.011	<u>0.287 ± 0.015</u>	0.399 ± 0.009
	0.20	**0.225 ± 0.008**	<u>0.244 ± 0.013</u>	0.342 ± 0.014	0.320 ± 0.017	0.341 ± 0.068	0.261 ± 0.008	0.245 ± 0.011	0.391 ± 0.010
	0.30	**0.212 ± 0.007**	0.240 ± 0.014	0.335 ± 0.015	0.300 ± 0.022	0.277 ± 0.059	0.236 ± 0.008	<u>0.221 ± 0.008</u>	0.384 ± 0.009
	0.50	0.194 ± 0.007	0.227 ± 0.012	0.340 ± 0.022	0.297 ± 0.016	**0.168 ± 0.003**	0.209 ± 0.008	<u>0.193 ± 0.008</u>	0.386 ± 0.009
Solar	0.05	**0.095 ± 0.015**	0.190 ± 0.020	0.241 ± 0.102	<u>0.115 ± 0.015</u>	0.374 ± 0.033	0.142 ± 0.016	0.165 ± 0.014	0.291 ± 0.009
	0.10	**0.083 ± 0.015**	0.159 ± 0.013	0.251 ± 0.081	<u>0.114 ± 0.014</u>	0.375 ± 0.038	0.124 ± 0.018	0.132 ± 0.015	0.276 ± 0.010
	0.20	**0.072 ± 0.015**	0.149 ± 0.020	0.314 ± 0.035	0.109 ± 0.016	0.217 ± 0.023	<u>0.108 ± 0.014</u>	0.109 ± 0.012	0.270 ± 0.010
	0.30	**0.061 ± 0.012**	0.135 ± 0.014	0.338 ± 0.05	0.108 ± 0.016	0.156 ± 0.002	0.100 ± 0.015	<u>0.098 ± 0.012</u>	0.266 ± 0.010
	0.50	**0.054 ± 0.013**	0.098 ± 0.013	0.315 ± 0.080	0.107 ± 0.015	<u>0.079 ± 0.011</u>	0.094 ± 0.013	0.088 ± 0.013	0.262 ± 0.009
Traffic	0.05	0.283 ± 0.016	**0.246 ± 0.010**	0.406 ± 0.074	0.318 ± 0.014	0.337 ± 0.045	0.293 ± 0.007	<u>0.261 ± 0.010</u>	0.363 ± 0.007
	0.10	**0.211 ± 0.012**	<u>0.214 ± 0.007</u>	0.319 ± 0.025	0.288 ± 0.018	0.288 ± 0.017	0.237 ± 0.006	0.245 ± 0.009	0.362 ± 0.006
	0.20	**0.168 ± 0.006**	0.216 ± 0.006	0.270 ± 0.012	0.271 ± 0.011	0.269 ± 0.017	<u>0.197 ± 0.005</u>	0.224 ± 0.008	0.361 ± 0.006
	0.30	**0.151 ± 0.007**	<u>0.172 ± 0.008</u>	0.251 ± 0.006	0.259 ± 0.012	0.240 ± 0.037	0.180 ± 0.006	0.197 ± 0.007	0.355 ± 0.006
	0.50	**0.139 ± 0.007**	0.171 ± 0.005	0.278 ± 0.040	0.240 ± 0.021	<u>0.144 ± 0.022</u>	0.160 ± 0.008	0.161 ± 0.060	0.354 ± 0.007
TimeFlow improvement		/	24.14 %	50.53 %	31.61 %	36.12 %	20.33 %	18.90 %	53.40 %

Qualitative results



Forecasting experiment

Setting



Quantitative results

Table 2: Mean MAE forecast results averaged over different time windows. Each time, the model is trained on one time window and tested on the others (there are 2 windows for *SolarH* and 5 for *Electricity* and *Traffic*). H stands for the horizon. Bold results are best, and underlined results are second best. TimeFlow improvement represents the overall percentage improvement achieved by TimeFlow compared to the specific method being considered.

		Continuous methods			Discrete methods			
	H	TimeFlow	DeepTime	Neural Process	Patch-TST	DLinear	AutoFormer	Informer
Electricity	96	<u>0.228 ± 0.028</u>	0.244 ± 0.026	0.392 ± 0.045	**0.221 ± 0.023**	0.241 ± 0.030	0.546 ± 0.277	0.603 ± 0.255
	192	<u>0.238 ± 0.020</u>	0.252 ± 0.019	0.401 ± 0.046	**0.229 ± 0.020**	0.252 ± 0.025	0.500 ± 0.190	0.690 ± 0.291
	336	<u>0.270 ± 0.031</u>	0.284 ± 0.034	0.434 ± 0.076	**0.251 ± 0.027**	0.288 ± 0.038	0.523 ± 0.188	0.736 ± 0.271
	720	<u>0.316 ± 0.055</u>	0.359 ± 0.051	0.607 ± 0.150	**0.297 ± 0.039**	0.365 ± 0.059	0.631 ± 0.237	0.746 ± 0.265
SolarH	96	**0.190 ± 0.013**	**0.190 ± 0.020**	0.221 ± 0.048	0.262 ± 0.070	0.208 ± 0.014	0.245 ± 0.045	0.248 ± 0.022
	192	**0.202 ± 0.020**	<u>0.204 ± 0.028</u>	0.244 ± 0.048	0.253 ± 0.051	0.217 ± 0.022	0.333 ± 0.107	0.270 ± 0.031
	336	<u>0.209 ± 0.017</u>	**0.199 ± 0.026**	0.240 ± 0.006	0.259 ± 0.071	0.217 ± 0.026	0.334 ± 0.079	0.328 ± 0.048
	720	**0.218 ± 0.041**	<u>0.229 ± 0.024</u>	0.403 ± 0.147	0.267 ± 0.064	0.249 ± 0.034	0.351 ± 0.055	0.337 ± 0.037
Traffic	96	<u>0.217 ± 0.032</u>	0.228 ± 0.032	0.283 ± 0.027	**0.203 ± 0.037**	0.228 ± 0.033	0.319 ± 0.059	0.372 ± 0.078
	192	<u>0.212 ± 0.028</u>	0.220 ± 0.022	0.292 ± 0.024	**0.197 ± 0.030**	0.221 ± 0.023	0.368 ± 0.057	0.511 ± 0.247
	336	<u>0.238 ± 0.034</u>	0.245 ± 0.038	0.305 ± 0.039	**0.222 ± 0.039**	0.250 ± 0.040	0.434 ± 0.061	0.561 ± 0.263
	720	<u>0.279 ± 0.050</u>	0.290 ± 0.052	0.339 ± 0.038	**0.269 ± 0.057**	0.300 ± 0.057	0.462 ± 0.062	0.638 ± 0.067
TimeFlow improvement		/	3.74 %	29.06 %	3.23 %	6.92 %	42.09 %	48.57 %