CAPE: CHANNEL-ATTENTION-BASED PDE PARAMETER EMBEDDINGS FOR SCIML - SUPPLEMENTARY MATERIAL

A ADDITIONAL RELATED WORK

Parameter Embedding There has been an interest to put additional information to DNN. For example, Transformer-type models take into account the information of the position of words in the sentence using positional encoding (Vaswani et al., 2017; Shaw et al., 2018; Huang et al., 2018). In the case of data generation, cGAN (Mirza & Osindero, 2014) accepts a conditional parameter to the generator network. In the case of SciML, PINN (Cai et al., 2022) and PINO (Li et al., 2021b) can explicitly take into account PDE parameters during training but cannot change them during the test time. Recently Message-passing PDE solver (Brandstetter et al., 2022) was proposed in which PDE parameters and boundary conditions can be freely embedded into the network. However, this is specialized only for these models, and cannot apply the other models as our proposed method.

B DETAILED TRAINING SETUP

B.1 GENERAL SETUP

As is explained in Sec. 3, we used datasets provided by PDEBench (Takamoto et al., 2022) a benchmark for SciML from which we downloaded datasets of the following PDEs: 1D Advection equation, 1D Burgers equation, and 2D compressible NS equations. For 1-dimensional PDEs, we used $N = 9000$ training instances and 1000 test instances for each PDE parameter with spatial resolution: 128 ($\Delta x = 1/128$) and temporal step-size: $\Delta t = 0.05$. For 2-dimensional NS equations, we used $N = 900$ training instances and 100 test instances for each PDE parameter with spatial resolution: 64×64 ($\Delta x = \Delta y = 1/128$) and temporal step-size: $\Delta t = 0.05$.

Concerning the training, the optimization was performed with Adam (Kingma & Ba) for 100 epochs. The learning rate was set as 3×10^{-3} which is divided by 2.0 every 20 epochs. The mini-batch size we used was 50 for all the cases. To stabilize the CAPE module's training in the initial phase, we empirically found it is a little better if we have a warm-up phase during which only CAPE module is updated. We performed warm-up for the first 3 epochs, which slightly reduce the final performance fluctuations resulting from the randomness of the initial weights of the network. In the CAPE module, the kernel size of the depth-wise convolution was set as: 5. In our experiments, we stacked 3 CAPE modules before providing the output with the BASE networks. Note that the channel parameter c of the second and 3rd network were set as: $2c$. The training was performed on GeForce RTX 2080 GPU for 1D PDEs and GeForce GTX 3090 for 2D NS equations. For PINO loss, we set the coefficient 1 following the original implementation.

The hyper-parameters and the BASE network parameters are listed in Tab. 5:

Dimension	Model	width	mode	d	mode (CAPE)	α
1D	FNO	36	12	–	–	–
	FNO w.t. CAPE	20	12	64	12	5.7×10^{-5}
2D	FNO	28	12	–	–	
	FNO w.t. CAPE	20	12	64	9	8.3×10^{-5}
Dimension	Model	init features		d	mode (CAPE)	α
1D	Unet	32		–	–	–
	Unet w.t. CAPE	32		64	12	5.7×10^{-5}
2D	Unet	32		–	–	
	Unet w.t. CAPE	30		64	9	8.3×10^{-5}

Table 5: FNO Network Parameters

B.1.1 NETWORKS' SIZES COMPARISON

The networks' structures of the BASE models are presented in Tab. 5, while the resulting network size is listed in Tab. 6.

Dimension	Model	# Parameters
1D	FNO	73K
	FNO w.t. CAPE	68K
	Unet	2.71M
	Unet w.t. CAPE	2.75M
2D	FNO	0.91M
	FNO w.t. CAPE	0.82M
	Unet	7.8M
	Unet w.t. CAPE	7.2M

Table 6: Model Size

B.2 CONDITIONAL MODELLING

Here we provide a more detailed explanation for the conditional models in Sec. 3. In this paper, the conditional models have the same model structures as the vanilla ones, but we only change the input data as:

$$u^k \in \mathbb{R}^{C\times N_1\times\cdots} \rightarrow \text{concatenate}(u^k, \lambda) \in \mathbb{R}^{(C+1)\times N_1\times\cdots}, \tag{17}$$

where the PDE parameter are taken as a part of the input by concatenating it to the field data's new channel dimension. Although it is possible to consider a more elaborate method, such as performing an MLP on the PDE parameters, we avoid those cases for simplicity.

B.3 MODIFICATION ON THE MESSAGE-PASSING PDE SOLVERS

In Sec. 3, we consider the Message-Passing PDE Solvers (Brandstetter et al., 2022) as a baseline model that accepts PDE parameters. For a fair comparison, we are forced to modify the model as (1) accepting only 1-time step data, (2) adding a case of "time-window" parameter with 10 for the decoder. Concerning the first case, the original model assumes to accept sequential data whose time-step size must be equal to the size of the "time-window" parameter. For the second modification, we added a new 1D-convolution layer accepting the "time-window" parameter equal to 10. The detailed structure of the new decoder is provided in Tab. 7.

Module	in-channel	out-channel	kernel size	stride
1D Conv-1	1	8	18	5
1D Conv-2	8	1	14	1

Table 7: Decoder CNN structure for Message-Passing PDE Solvers

C DISCUSSION OF RESULTS FOR THE PINO LOSS

Reason why PINO loss does not work The PINO loss function is an emulation of the PINO loss function using ML's output. For example, the PINO loss function of the 1D Advection equation case is:

$$L_{\text{PINO}} = \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} - \beta\mathcal{F}^{-1}(ik\mathcal{F}(u)). \tag{18}$$

On the other hand, the usual spectral method solves the equation as:

$$\tilde{u}_j^{n+1} = u_j^{n-1} + 2\Delta t\beta\mathcal{F}^{-1}(ik\mathcal{F}(u)). \tag{19}$$

By substituting Eq. 19, Eq. 18 reduces to:

$$L_{\text{PINO}} = \frac{u_j^{n+1} - \tilde{u}_j^{n+1}}{2\Delta t}. \tag{20}$$

This shows that the PINO loss function penalizes the machine learning model prediction to be close to the spectral method prediction. However, in general, the classical direct simulation methods have to use the time-step size Δt restricted by the theoretical stability condition, such as the CFL condition. And the prediction $\tilde{u}_j^{n+1}$ becomes completely wrong if the used Δt does not satisfy the stability condition, resulting in the PINO loss function leading to a completely harmful effect for the ML models. In our experiments, $\Delta t = 0.05$ is larger than the time step demanded by the stability condition, e.g., $\Delta t < \Delta x/\beta \sim 0.0025$ if we set $\Delta x = 1/128, \beta = 1$. So, we consider that our experiment result showing worse error from PINO loss function is a natural result from this consideration.

D Detailed Ablation Study Results

D.1 PDE Parameter Dependence Study

PDE	β type	0.1	0.2	0.4	0.7	1.0	2.0	4.0	7.0
Advection	Base	0.716	0.700	0.638	0.680	0.714	0.721	0.700	0.729
	Cape	0.846	0.056	0.040	0.035	1.218	0.046	0.049	3.300

PDE	ν type	0.001	0.002	0.004	0.007	0.01	0.02	0.04	0.07	0.1
Burgers	Base	0.223	0.216	0.218	0.201	0.198	0.173	0.138	0.134	0.124
	Cape	0.185	0.179	0.167	0.155	0.155	0.138	0.127	0.106	0.107

PDE	ν type	0.2	0.4	0.7	1.0	2.0	4.0
Burgers	Base	0.168	0.335	0.458	0.674	1.626	2.460
	Cape	0.081	0.094	0.113	0.104	0.079	0.253

D.2 Curriculum Strategy Study

E Detailed Description of the Curriculum Strategy

Fig. 6 plots the profile of Eq. 12 in terms of the epoch number where the maximum epoch number is assumed 100. We also provided the detailed algorithm of our curriculum strategy in terms of epochs and temporal steps in Algorithm 1. In our all the calculation with curriculum strategy, we set: $\Delta = 0.2$.

F Ablation Study for CAPE module structure

In this section, we provided results of ablation study for our CAPE module's internal structure to provide an insight of the inductive bias of CAPE. In this study, we performed training without (1) spectral-convolution, (2) 1×1-convolution, and (3) depthwise-convolution. The results were provided in Tab. 9 that indicates that all the 3-convolution layers and LayerNormalization play important roles on the error, but the spectral-convolution has the strongest impact. However, it also shows that the important factor depends on PDEs because of the difference of PDE natures (e.g., advection, diffusion, or non-linear system equations, and so on). It also indicates that our selection always shows a better result, though not always the best.

PDE	$\eta = \zeta$ type	10^{-8}	0.001	0.004	0.007	0.01	0.04	0.07	0.1
2D NS ($M = 0.1$)	BASE	0.508	0.500	0.488	0.491	0.529	1.447	3.132	5.228
	CAPE	0.516	0.487	0.482	0.462	0.486	0.965	1.692	2.582
2D NS ($M = 1.0$)	BASE	0.579	0.545	0.495	0.471	0.453	0.635	1.141	1.962
	CAPE	0.569	0.544	0.501	0.485	0.474	0.494	0.585	0.779

PDE	Model	Ablation	nMSE	
Advection	FNO	curriculum strategy	**0.04**	
		pure Autoregressive	0.11	(+**0.07**)
		pure Teacher-Forcing	**0.04**	(±0.00)
Advection	Unet	curriculum strategy	0.11	
		pure Autoregressive	0.21	(+**0.10**)
		pure Teacher-Forcing	**0.10**	(−0.01)
Burgers	FNO	curriculum strategy	**0.13**	
		pure Autoregressive	0.16	(+**0.03**)
		pure Teacher-Forcing	**0.13**	(+0.00)
Burgers	Unet	curriculum strategy	**0.45**	
		pure Autoregressive	0.92	(+**0.47**)
		pure Teacher-Forcing	0.75	(+0.30)
2D NS	FNO	curriculum strategy	$\mathbf{8.0 \times 10^{-1}}$	
		pure Autoregressive	$1.3 \times 10^{+0}$	(+0.5)
		pure Teacher-Forcing	$3.2 \times 10^{+0}$	(+**2.4**)
2D NS	Unet	curriculum strategy	$\mathbf{7.0 \times 10^{-1}}$	
		pure Autoregressive	$1.0 \times 10^{+0}$	(+**0.3**)
		pure Teacher-Forcing	$1.0 \times 10^{+0}$	(+**0.3**)

Table 8: Ablation study for the Advection, Burgers and 2D CFD equations with FNO as BASE model.

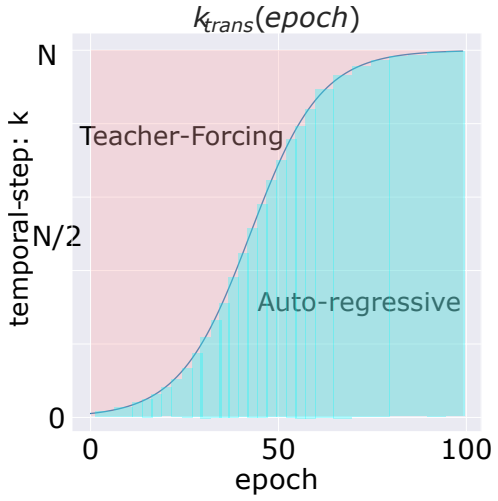

Figure 6: A plot of an instance of the function defined in Eq. 12 where we set: $\Delta = 0.25$.

Algorithm 1 Algorithm of the curriculum training strategy

Input model parameters $\boldsymbol{\theta}$, training epoch number n, total training epoch number M, Training samples: $\{\boldsymbol{u}^i\}_{i=0,\cdots,N}$, temporal index k, final time step of the training sample N, ϵ is the random noise.

for n $= 0$ **to** M **do**
  Calculate k_{trans} following Eq. 12,
  for k $= 0$ **to** $N - 1$ **do**
    if $k \leq k_{\text{trans}}$ **then**
      $\tilde{\boldsymbol{u}}^{k+1} = \text{NN}(\tilde{\boldsymbol{u}}^k + \epsilon; \boldsymbol{\theta})$
    else
      $\tilde{\boldsymbol{u}}^{k+1} = \text{NN}(\boldsymbol{u}^k + \epsilon; \boldsymbol{\theta})$
    end if
    $\mathbf{L}^k \leftarrow \text{MSE}(\tilde{\boldsymbol{u}}^{k+1}, \boldsymbol{u}^{k+1})$
  end for
  $\boldsymbol{\theta} \leftarrow$ Optimizer $(\sum_{k=1}^{N} \mathbf{L}^k)$
end for

PDE	model	Ablation	nRMSE
1D Advection	FNO	BASE	0.04
		w/t Spectral Convolution	0.06 **(+0.02)**
		w/t 1×1 Convolution	0.05 (+0.01)
		w/t Depthwise Convolution	0.05 (+0.01)
		w/t LN	0.03 (-0.01)
1D Burgers	FNO	BASE	0.13
		w/t Spectral Convolution	0.12 (-0.01)
		w/t 1×1 Convolution	0.13 (+0.00)
		w/t Depthwise Convolution	0.13 (+0.00)
		w/t LN	0.08 (-0.05)
2D CFD	FNO	BASE	0.80
		w/t Spectral Convolution	1.03 (+0.23)
		w/t 1×1 Convolution	0.69 (-0.11)
		w/t Depthwise Convolution	0.80 (+0.00)
		w/t LN	1.25 **(+0.45)**

Table 9: Ablation study of CAPE internal Structure.