Spectrally Normalized Adaptive Neural Identifier for Dynamic Modeling and Trajectory Tracking Control of Unmanned Aerial Vehicle

Shaofeng Chen[1], Yu Kang[1,2,3(✉)], Yunbo Zhao[1], and Yang Cao[1]

[1] Department of Automation, University of Science and Technology of China, Hefei 230026, China
csf123@mail.ustc.edu.cn, {kangduyu,ybzhao,forrest}@ustc.edu.cn

[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China

[3] Institute of Advanced Technology, University of Science and Technology of China, Hefei 230088, China

Abstract. Accurate dynamic modeling is difficult for aerobatic unmanned aerial vehicles flying at their physical limit, due to the model uncertainty caused by unobservable hidden states like airflow and vibrations. Although some progresses have been made, these hidden states are still not properly characterized, rendering system identification problem for aerobatic unmanned aerial vehicle extremely challenging. To address this issue, a novel spectrally normalized adaptive neural identifier is proposed for the dynamic modeling of aerobatic unmanned aerial vehicles. Specifically, to characterize the model uncertainty, we propose a spectrally normalized adaptive neural network (SNANet) to extract deep features representing the hidden states of the system. Particularly, the proposed SNANet adopts a multi-model adaptive structure, quickly and dynamically updating the model online. Furthermore, the spectral normalization constraint is introduced into the training process to ensure the Lipschitz stability of the SNANet. Consequently, a trajectory tracking control scheme including the sliding mode controller and SNANet is presented. The modeling effectiveness of the proposed method is verified on a real flight dataset. The results demonstrate that our method has high modeling accuracy, short training time, and fast model response speed.

Keywords: Spectral normalization · Model uncertainty · Adaptive neural identifier · Aerobatic unmanned aerial vehicle

1 Introduction

Unmanned aerial vehicles (UAVs) achieve vertical take-off and landing, coordinated steering, and other flight tasks in a small space, due to their high flexibility and strong adaptability, so they have been widely applied in military and

L. Yan et al. (Eds.): ICGNC 2022, LNEE 845, pp. 6998–7007, 2023.
https://doi.org/10.1007/978-981-19-6613-2_674

civil fields. However, accurately modeling aerobatic UAVs remains a challenging research problem due to the need to capture dynamic couplings such as engine dynamics, aerodynamics, etc. Dynamic coupling is difficult to model because it relies on hidden states that are difficult to measure such as airflow and vibration.

The UAV system modeling is well studied. The simple linear model constructed by the mechanism modeling method is only suitable for the controller design of simple trajectory. More complex nonlinear models built by the grey-box modeling method enable some simple aerobatics [1]. Researches [2] suggest that the key to aerobatic UAV modeling lies in unobserved state variables. Although hidden states such as airflow cannot be directly observed, they can be deduced from measurement sequences of other observable states [2]. Traditional methods usually regard the identification of hidden states as a parameter learning problem, but these methods all require prior knowledge and assumptions, which limit the representation ability of the model.

To model uncertainty caused by hidden states without being overly-constrained by the aforementioned methods, scholars take deep neural networks (DNNs) to mine the hidden states. Research [2] regarded the UAV dynamics modeling as a high-dimensional regression problem and used a fully connected neural network to learn the system model. In [3], a convolutional neural network was used to characterize the uncertainty of the helicopter system and based on this, a backstepping adaptive controller was designed. Research [4] used a temporal convolutional neural network to model the aerodynamics of the UAV and combined it with first-principles modeling methods to build a system model.

Although these methods have achieved good performance, the following three issues remain unresolved: 1) It is usually hard to analyze DNNs, making the analysis of stability with DNN-based controllers difficult; 2) The UAV is dynamic and online, and the requirement on the speed of DNN is very high; 3) DNN is high-dimensional and may generate unpredictable outputs, which in turn lead to instability of the feedback control loop.

To overcome the above issues, we propose a novel spectrally normalized adaptive neural identifier for the dynamic modeling of aerobatic UAVs. Specifically, we propose a SNANet to characterize the model uncertainty caused by hidden states. The proposed SNANet adopts a multi-model adaptive structure and introduces spectral normalization constraints into the training process, which ensures the fast and dynamic online update capability and the Lipschitz stability. Then, the model uncertainty described by the SNANet is combined with the traditional dynamics model to achieve a complete characterization of the UAV. Finally, using the Lipschitz property of the SNANet, a trajectory tracking controller is designed and the stability of the system is proved.

The main contributions are summarized as follows.

1) We propose a novel SNANet with a multi-model adaptive structure to characterize the model uncertainty, making the model both a prominent modeling accuracy and a fast and dynamic online update capability.

2) We introduce a spectral normalization for the training process of the proposed SNANet, enhancing the Lipschitz stability of the SNANet in a learning-theoretic sense.
3) Based on the SNANet-based model, we propose a trajectory tracking controller via the sliding mode control method. The Lyapunov theory is adopted to prove the stability of the control scheme.

2 Problem Formulation

The model of the UAV is given as:

$$\dot{r} = v \tag{1}$$

$$m\dot{v} = mg + Rh_\tau + h_d \tag{2}$$

where $r \in \mathbb{R}^3$ represents the position, $v \in \mathbb{R}^3$ denotes the airframe velocity, m and $R \in SO(3)$ present respectively the mass and attitude rotation matrix of the UAV, and $h_\tau = [0, 0, T]^{\mathrm{T}}$ is the total thrust force. The difficulty of accurate dynamic modeling of the aerobatic UAV lies in the description of the uncertainty term $h_d = [h_{dx}, h_{dy}, h_{dz}]^{\mathrm{T}}$ caused by hidden states like airflow and vibrations.

Our Objectives: 1) Design a DNN-based method to describe the uncertainty terms h_d. Because the UAV system is dynamic and online, the designed DNN is required to have high identification accuracy, low computational burden, and fast identification speed; 2) Design a trajectory tracking controller based on the proposed DNN for validating the effectiveness of the proposed modeling approach in a control task. As DNNs may produce unpredictable outputs, the proposed DNN must be analyzable.

3 Spectrally Normalized Adaptive Neural Network

This section proposes a novel SNANet to learn the uncertainty term h_d in the UAV model. First, the structure of SNANet is described in detail. Then, the adaptive updating rule for convex coefficients is given, and a performance analysis of convex coefficients of the SNANet is carried out. Finally, the training method of SNANet constrained by spectral normalization is given.

Network Structure: To fulfil the requirements of the UAV control system, a SNANet is designed in terms of computational burden, identification accuracy, and speed. The structure of the SNANet is given in Fig. 1. It uses a multi-model adaptive structure where m deep feature learning networks (DFNets) $\mathcal{D}_i(i = 1, 2, \cdots, m)$ with the same structure are connected by convex coefficients $\lambda_i \in [0, 1]$. The SNANet updates the λ_i rather than the DFNet weights during the system online identification process, so the convergence speed of the proposed SNANet can be greatly improved.

The SNANet satisfies the following three properties: 1) The initial state of SNANet is equal to the convex combination of all DFNets, i.e.

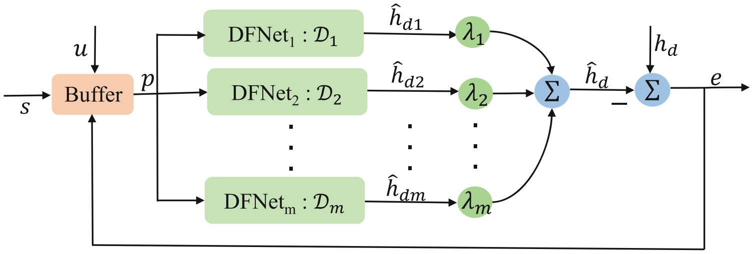


Fig. 1. Structure of the proposed SNSNet

$\sum_{i=1}^{m} \varphi_i(0)\mathcal{D}_i(0) = \mathcal{D}(0)$; 2) The SNANet needs to meet the criterion for convexity, i.e. $\sum_{i=1}^{m} \lambda_i = 1$; 3) The SNANet should converge at the same rate as a single DFNet.

Considering that the DNN using the ReLU activation function can better suppress the vanishing gradient problem, DFNet uses this activation function to learn h_d of the system, which can be expressed as:

$$\mathcal{D}_i(p_i, \vartheta_i) = W_i^{L+1}\kappa\left(W_i^L\left(\kappa\left(W_i^{L-1}\left(\cdots\kappa\left(W_i^1 p_i\right)\cdots\right)\right)\right)\right) \tag{3}$$

where p_i is the input of the network, $\vartheta_i = W_i^1, W_i^2, \cdots, W_i^{L+1}$ is the network weight coefficient, κ is the ReLU activation function.

Adaptive Updating Rules for Convex Coefficients: The output of SNANet is a convex combination of m DFNets, as follows:

$$\hat{h}_d(t) = \lambda_1(t)\hat{h}_{d1}(t) + \cdots + \lambda_m(t)\hat{h}_{dm}(t) \tag{4}$$

The identification error for SNANet is:

$$e(t) = h_d(t) - \hat{h}_d(t) \tag{5}$$

Combining the Eq. (4)–(5) and the criterion for convexity:

$$\begin{aligned} e(t) &= h_d(t) - \left(\lambda_1(t)\hat{h}_{d1}(t) + \cdots + \lambda_m(t)\hat{h}_{dm}(t)\right) \\ &= \sum_{i=1}^{m-1} \lambda_i(t)\tilde{e}_i(t) + e_m(t) \end{aligned} \tag{6}$$

where $e_i(t) = \hat{h}_{di}(t) - h_d(t)$ and $\tilde{e}_i(t) \triangleq e_i(t) - e_m(t)$. The Eq. (6) can be further rewritten as:

$$\tilde{e}(t) = \tilde{\mathcal{G}}^T(t)\tilde{\lambda}(t) \tag{7}$$

where, $\tilde{\mathcal{G}}(t) = [\tilde{e}_1(t), \cdots, \tilde{e}_{m-1}(t)] \in \mathbb{R}^{m-1}$, $\tilde{\lambda}(t) = [\lambda_1(t), \cdots, \lambda_{m-1}(k)] \in \mathbb{R}^{m-1}$.

The adaptive updating rule for the convex coefficients can be designed as follows:

$$\tilde{\lambda}(t) = \tilde{\mathcal{G}}\tilde{e}(t-1) - \tilde{\mathcal{G}}\tilde{\mathcal{G}}^T\tilde{\lambda}(t-1) + \tilde{\lambda}(t-1) \tag{8}$$

So, the system identification error can be described as a convex combination of the modeling errors of each DFNet, which can be expressed as follows:

$$e = \sum_{i=1}^{m} \lambda_i \mathcal{D}_i - h_d = \sum_{i=1}^{m} \lambda_i e_i \tag{9}$$

Training Under Spectral Normalization Constraints: To obtain a stable SNANet, the optimization objectiveness is to minimize the prediction error subject to a constrained Lipschitz constant, which can be mathematically described as:

$$\begin{aligned} &\underset{\vartheta}{\text{minimize}} && \sum_{t=1}^{T} \frac{1}{T} \left\| h_d^* - \mathcal{D}(p, \vartheta) \right\|_2 \\ &\text{subject to} && \|\mathcal{D}\|_{\text{Lip}} \le \delta \end{aligned} \tag{10}$$

where, h_d^* is the actual uncertainty of the UAV and p is the observed state, and ϑ is the parameter to be learned. Research [5] suggests that training a SNANet with the bounded Lipschitz constants is a hallmark of machine learning stability.

4 Trajectory Tracking Control Based on SNANet

Trajectory Tracking Controller: To design the controller, the following composite variables are introduced:

$$c = \dot{\tilde{r}} + \mathit{\Gamma} \tilde{r}. \tag{11}$$

where, $\tilde{r} = r - r_{\mathfrak{e}}$, $\mathit{\Gamma}$ is a diagonal or positive definite matrix. Then the controller needs to be designed such that $\tilde{r}(t)$ converges exponentially to 0 on the manifold determined by $c = 0$. Let $v_{\mathfrak{d}} = \dot{r}_{\mathfrak{e}} - \mathit{\Gamma}\tilde{r}$, Eq. (11) can be rewritten as:

$$c = \dot{r} - v_{\mathfrak{d}} \tag{12}$$

where, $v_{\mathfrak{d}}$ denotes the reference velocity. Using the method proposed in Sect. 3, $\hat{h}_d(p)$ is used to approximate the model uncertainty h_d and p is the input to the SNANet. The desired total thrust force $h_{\mathfrak{e}}$ is defined as follows:

$$h_{\mathfrak{e}} = (Rh_{\mathcal{T}})_{\mathfrak{e}} = \bar{h}_{\mathfrak{e}} - \hat{h}_d \tag{13}$$

where $\bar{h}_{\mathfrak{e}} = m\dot{v}_{\mathfrak{d}} - A_v c - mg$. Substituting Eq. (13) into Eq. (2):

$$m\dot{c} + A_v c = \epsilon \tag{14}$$

where, $\epsilon = h_d - \hat{h}_d$. Thus, $\tilde{r}(t) \to 0$ is bounded by global and exponential error as long as $\|\epsilon\|$ is bounded.

Stability Analysis: The stability analysis is given as following.

Theorem 1. *Given a time-varying $r_{\mathfrak{e}}(t)$, using the designed controller (13) that satisfies $\lambda_{\min}(A_v) > L_d\alpha$ ensures that the error of the composite variable c converges exponentially to $\lim_{t\to\infty}\|c(t)\| = \epsilon_m(\lambda_{\min}(A_v) - L_d\alpha)$ at a rate of $(\lambda_{\min}(A_v) - L_d\alpha)/m$, and $\tilde{r}$ exponentially converges, i.e.*

$$\lim_{t\to\infty}\|\tilde{r}(t)\| = \frac{\epsilon_m}{\lambda_{\min}(\mathit{\Gamma})\left(\lambda_{\min}(A_v) - L_d\alpha\right)} \tag{15}$$

Proof. We choose the Lyapunov function as follows:

$$V(c) = \frac{1}{2}m\|c\|^2 \tag{16}$$

By involving the designed controller (13) and universal approximation theorem, we obtain:

$$\dot{V} = c^{\mathrm{T}}\left(-A_v c + h_d - \hat{h}_d\right) \le -c^{\mathrm{T}}A_v c + \|c\|^2\epsilon_m \tag{17}$$

where ϵ_m is the upper bound on the learning error of $\hat{h}_d$. Let $\lambda_{\min}(A_v)$ denote the smallest eigenvalue of the positive definite matrix A_v. Using the Lipschitz property of $\hat{h}_d$ obtains:

$$\dot{V} = \sqrt{\frac{2V}{M}}\epsilon_m - \frac{2(\lambda - L_d\alpha)}{m}V \tag{18}$$

According to the comparison lemma, we have

$$\|c(t)\| \le \|c(t_0)\|\exp\left(-\frac{\lambda - L_d\alpha}{m}(t - t_0)\right) + \frac{\epsilon_m}{\lambda - L_d\alpha} \tag{19}$$

The results demonstrate finite-gain $\mathcal{L}_p$ stability and input state stability. Further, the graded combination of s and $\tilde{r}$ in Eq. (12) makes $\lim_{t\to\infty}\|\tilde{r}(t)\| = \lim_{t\to\infty}\|c(t)\|/\lambda_{\min}(\mathit{\Gamma})$, yielding (15).

5 Simulation Experiments

This section introduces the experimental settings and gives the related experimental results analysis, specifically: the training optimization method of the network, the introduction of comparison methods, model uncertainty characterization performance, and the performance of the proposed trajectory tracking control method.

5.1 Experimental Setup

Network Optimization: The flight dataset is from the Stanford Unmanned Aerial Vehicle Group, which employs professional pilots to control UAV to repeatedly complete 20 types of advanced aerobatic maneuvers, and to collect related flight data. The trajectories are recorded with a sampling interval of

Table 1. Identification errors for model uncertainty

	RBF	Deep ReLU	GRU	SNANet
RMSE	2.12	0.72	0.59	**0.46**

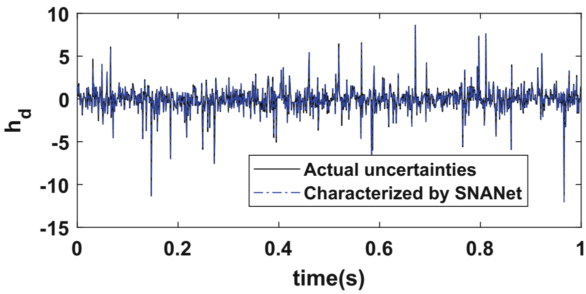


Fig. 2. Identification of model uncertainty by the proposed SNANet

0.01s. We randomly select 60%, 20%, and 20% data of as training, testing and validation sets, respectively. To prevent the results of a single test from being too one-sided, an 8-fold cross-validation experiment is adopted. The initial learning rate of SNANet is 0.001, the minimum learning rate is 10^{-6}, and the dropout size is 0.05. The learning rate decreases when the loss function is not improved for 10 consecutive epoch validation sets.

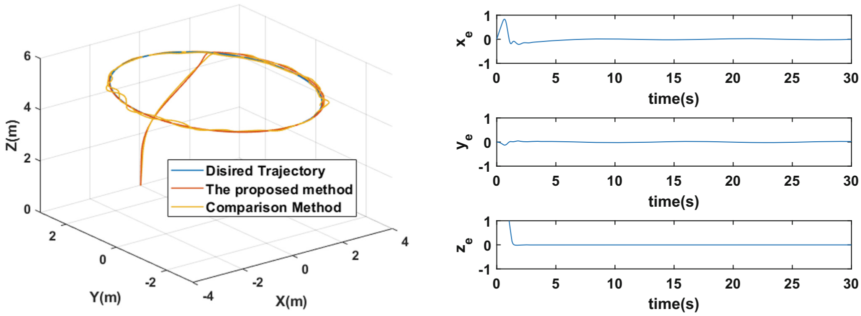
Comparison Methods: To fully prove the validity of the SNANet, we choose the RBF approach [6], the deep ReLU approach [7], and the GRU approach [8] for comparison. The RBF approach is one of the most widely used and effective traditional neural networks (NNs). The deep ReLU approach is the first DNN applied in the field of UAV system identification. The reason for choosing the GRU approach for comparison is that the most relevant DNNs for system identification are recurrent neural networks and their extensions, and GRU approach is a new type of these networks. To verify the performance of the proposed controller based on the SNANet, we choose the method [9] for comparison.

5.2 Performance

Higher Identification Accuracy: The uncertainty identification errors of the SNANet and the three comparison methods are shown in Table 1. Obviously, the uncertainty identification error of the proposed SNANet is the smallest. Compared with the three comparison methods, the uncertainty identification error is reduced by 78.30%, 36.11%, and 22.03% respectively. This is mainly because the RBF neural network only has the ability of simple nonlinear fitting, but can not mine the hidden states of the system. This is because SNANet is able to extract deep spatiotemporal features representing the hidden states and intrinsic laws of the system, so it has better uncertainty fitting ability.

Table 2. Time consumption comparison of different methods

	RBF	Deep ReLU	GRU	**SNANet**
Training time (min)	22.93	10.63	11.07	**3.96**
Predicting time (ms)	1.57	0.86	1.07	**0.45**



(a) (b)

Fig. 3. Circular trajectory tracking: (a) Dynamic trajectory; (b) Position error

Meanwhile, to more vividly demonstrate the model uncertainty identification ability of SNANet, Fig. 2 shows the identification results of model uncertainty by SNANet. As we can see, SNANet is able to accurately identify the model uncertainty if it changes drastically due to airflow, vibration, etc. The above experimental results fully prove that the SNANet has higher identification accuracy.

Faster Convergence Rate and Prediction Speed: Table 2 presents the comparison results of the training time and single prediction time. It is obvious that SNANet not only has the fastest convergence speed but also has the shortest single prediction time. Compared with the three comparison methods, the training time of SNANet is reduced by 85.55%, 63.69%, and 65.13% respectively, and the single prediction time is reduced by 71.34%, 47.67% and 57.94% respectively. This is because SNANet adopts a multi-model adaptive architecture. When the SNANet is applied online, it is not the network weights but the convex coefficients that are updated. The above results fully demonstrate that the proposed SNANet has faster convergence and prediction speed.

Better Trajectory Tracking Performance: The performance of the presented method was demonstrated through circular and switched trajectory tracking. Firstly, the circular tracking trajectory results of the proposed controller and the comparison method [9] are presented in Fig. 3(a). It presents that our method can still track the desired trajectory well with large model uncertainties, while the comparison method has a large error. To better show the control precision of our method, the tracking error of our method is given in Fig. 3(b). Our method can make the error converge to an arbitrarily small range of the origin.

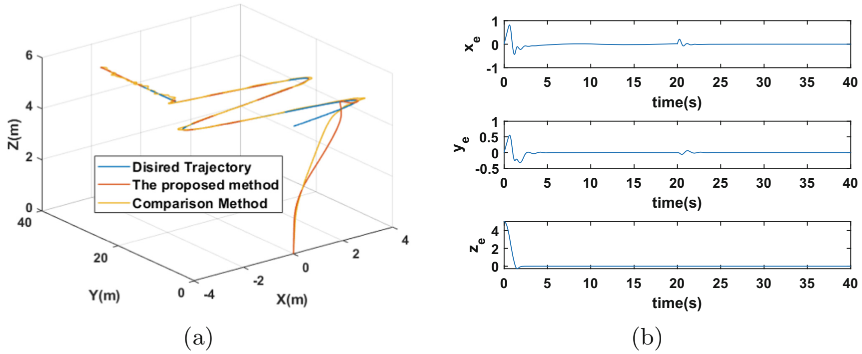


Fig. 4. Switched trajectory tracking: (a) Dynamic trajectory; (b) Position error

Furthermore, to validate the control performance of our method, we add uncertainty never used in training during trajectory tracking, and switch the expected trajectory type abruptly. The expected trajectory is a sin curve from 0 s to 20 s and suddenly switches to a straight line at 20 s. As presented in Fig. 4(a), the proposed method has better tracking performance, while the comparison method not only has a large tracking error but also cannot efficiently track the desired trajectory with sudden changes. Then, the position error during the trajectory switching tracking is presented in Fig. 4(b). The error of our controller can converge to an arbitrarily small range of the origin.

6 Conclusion

This paper proposes a novel spectrally normalized adaptive neural identifier for dynamic modeling and trajectory tracking of aerobatic UAVs. Firstly, to characterize model uncertainty, a novel SNANet is proposed. Due to the multi-model adaptive structure of the SNANet, it has a fast training speed and model response speed, which can meet the requirements of the UAV system dynamic online. Due to the superior hidden feature mining ability of the SNANet, it has a high accuracy of model uncertainty. Due to the spectral normalization constraint used in the training process, it has the Lipschitz property, which makes SNANet available for theoretical analysis. Furthermore, a complete system model is constructed by combining SNANet with the traditional UAV dynamics model. Finally, based on this model, the trajectory tracking controller of the UAV is designed, and the stability of the system is analyzed.

Acknowledgment. This work was supported by the Key Research and Development Project of China's Ministry of Science and Technology (2018AAA0100800).

References

1. Gavrilets, V., Martinos, I., Mettler, B., Feron, E.: Control logic for automated aerobatic flight of a miniature helicopter. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, p. 4834 (2002)
2. Punjani, A., Abbeel, P.: Deep learning helicopter dynamics models. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 3223–3230. IEEE (2015)
3. Kang, Y., Chen, S., Wang, X., Cao, Y.: Deep convolutional identifier for dynamic modeling and adaptive control of unmanned helicopter. IEEE Trans. Neural Networks Learn. Syst. **30**(2), 524–538 (2018)
4. Bauersfeld, L., Kaufmann, E., Foehn, P., Sun, S., Scaramuzza, D.: NeuroBEM: hybrid aerodynamic quadrotor model. ArXiv Preprint ArXiv:2106.08015 (2021)
5. Bartlett, P.L., Foster, D.J., Telgarsky, M.J.: Spectrally-normalized margin bounds for neural networks. Adv. Neural Inf. Process. Syst. **30** (2017)
6. Zhang, L., et al.: State-of-charge estimation of lithium-ion battery pack based on improved RBF neural networks. Complexity 2020 (2020)
7. Agarap, A.F.: Deep learning using rectified linear units (RELU). ArXiv Preprint ArXiv:1803.08375 (2018)
8. Heng, Y., et al.: A pan-specific GRU-based recurrent neural network for predicting HLA-I-binding peptides. ACS Omega **5**(29), 18321–18330 (2020)
9. Tian, Z., Zhou, Y.: Adaptive dynamic surface control of UAV based on RBF neural network. In: 2021 40th Chinese Control Conference, pp. 694–699. IEEE (2021)