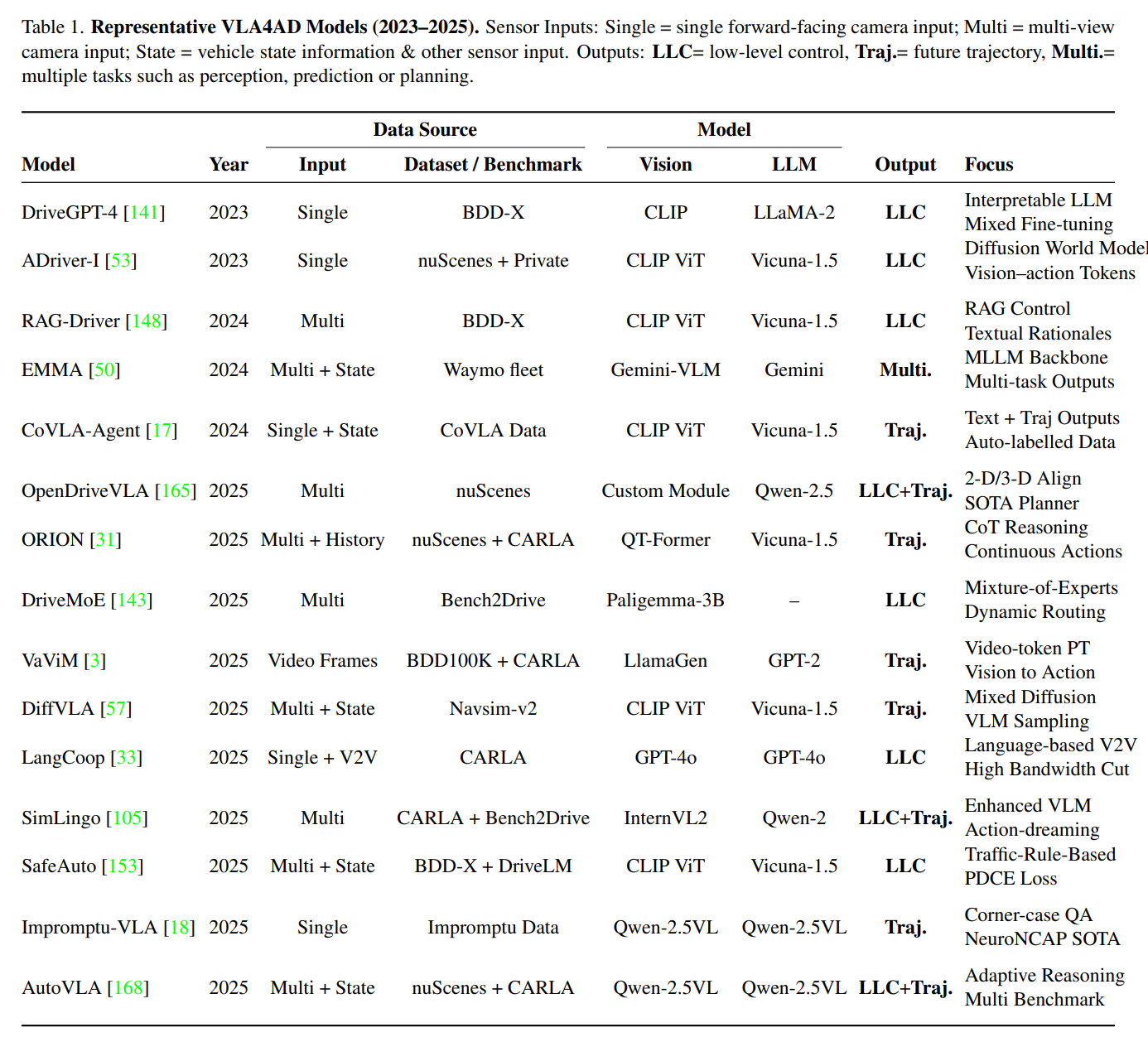
**A Survey on Vision-Language-Action Models for Autonomous Driving**

原文出处：<https://arxiv.org/pdf/2506.24044>

日期：2025.06



**ORION: A Holistic End-to-End Autonomous Driving Framework by Vision-Language Instructed Action Generation**

原文出处：<https://arxiv.org/abs/2503.19755>

日期：2025.03

源码：<https://xiaomi-mlab.github.io/Orion/>

**Motivation:**

（Abstract）

End-to-end (E2E) autonomous driving methods still struggle to make correct decisions in interactive closed-loop evaluation due to limited causal reasoning capability. Current methods attempt to leverage the Vision-Language Models (VLMs) to resolve this dilemma. However, the problem is still open that few VLMs for E2E methods perform well in the closed-loop evaluation due to the gap between the semantic reasoning space and the purely numerical trajectory output in the action space.

（Introduction）

leveraging VLMs for E2E autonomous driving is not trivial, as the capabilities of VLMs focus on the semantic reasoning space, while E2E methods only need the numerical planning results in the action space.

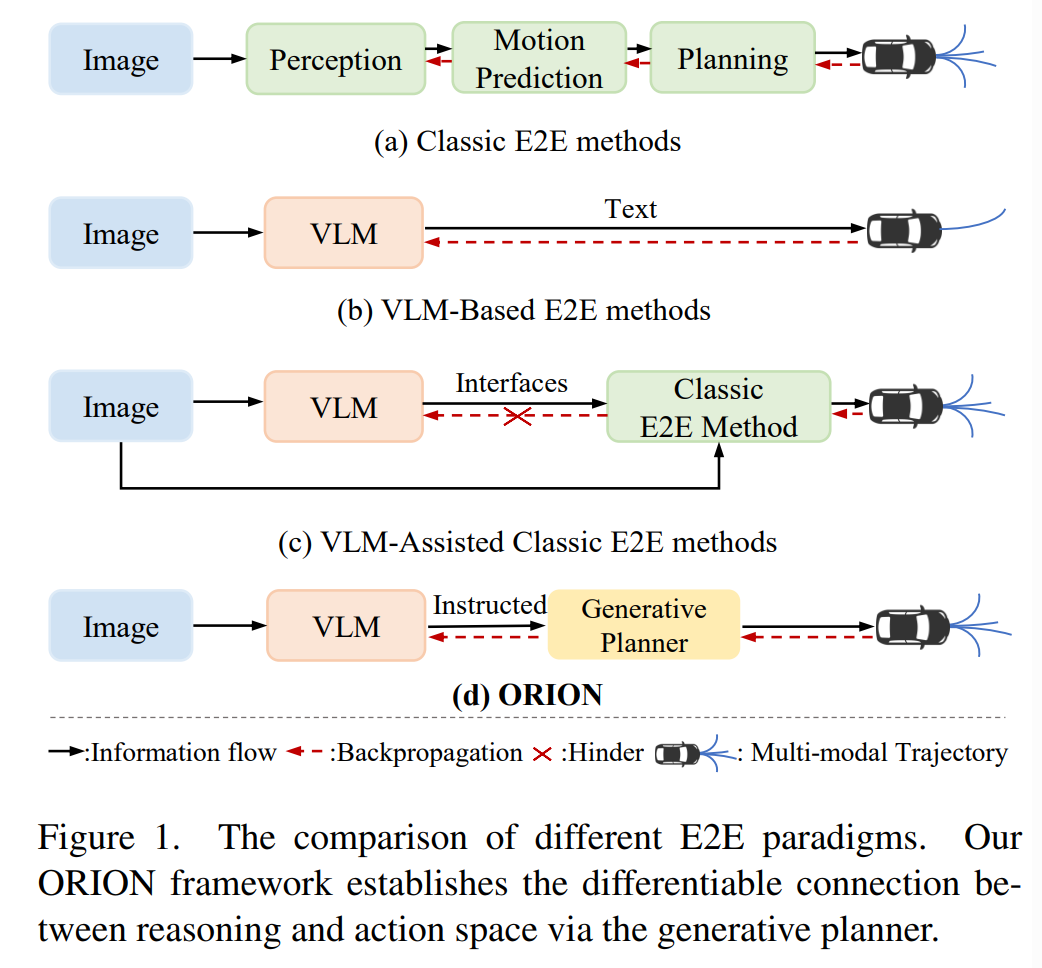


Fig. 1(b) and Fig. 1(c) adopt a carefully crafted interface to transmit the reasoning space information into the action space. However, they decouple these two spaces, hindering collaborative optimization between the trajectory optimization and the VLM reasoning process.

**Contribution:**

（Abstract）

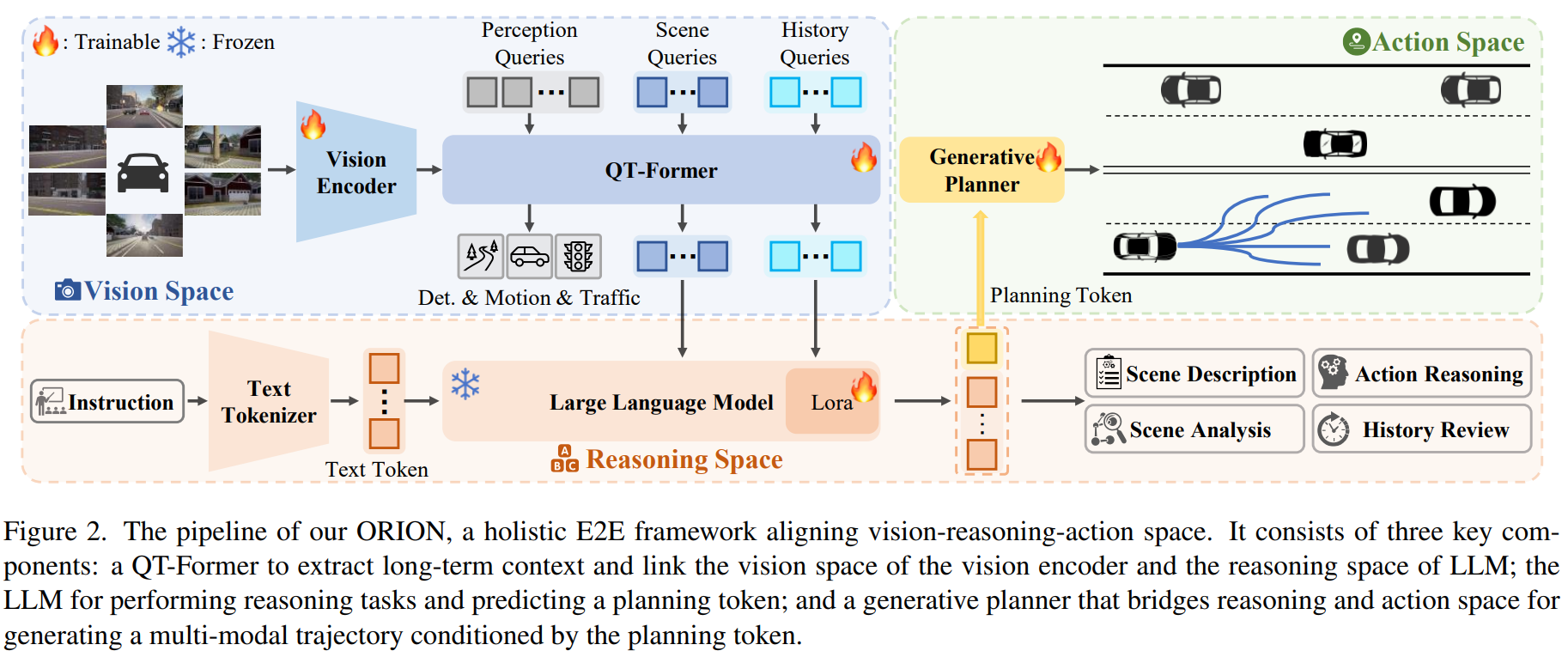
ORION uniquely combines a QT-Former to aggregate long-term history context, a LLM for driving scenario reasoning, and a generative planner for precision trajectory prediction. ORION further aligns the reasoning space and the action space to implement a unified E2E optimization for both visual question-answering (VQA) and planning tasks.

（Introduction）

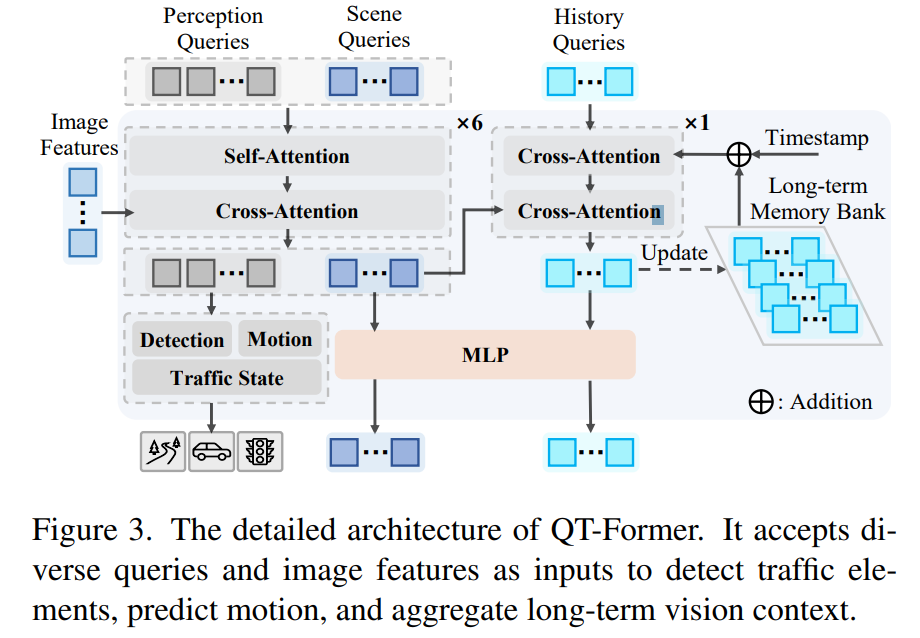
Considering that the reasoning space of VLM and the action space of trajectory belong to different domains, we introduce a generative planner to establish a unified latent representation for aligning the two spaces. With the help of the introduced module, we take advantage of VLMs’ reasoning information to construct trajectory.

It is well-known that long-term memory is necessary for E2E autonomous driving since historical information often influences trajectory planning within the current scene. Existing VLMs for E2E methods [19, 62] typically concatenate multi-frame images for temporal modeling. They are constrained by the token length of VLM and incur significant computational overhead. To address this issue, we introduce QT-Former (a query-based temporal module) to extract features.

**Method:**

**(1) QT-Former**

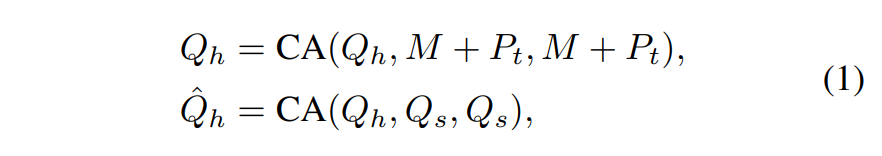
We first set up two types of learnable queries, the scene queries *Qs*∈ℝ*Ns×Cq* and the perception queries *Qp*∈ℝ*Np*×*Cq*, where *Ns* and *Np* are the number of scene and perception queries, respectively, and *Cq* is the channel of queries.



*Qs*, *Qp* are processed through self-attention (SA) to exchange their information. Then they interact with image features *Fm* with 3D positional encoding [37] *Pm* in the cross-attention (CA) module.

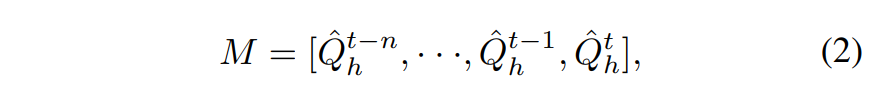
After that, the perception queries are fed into the multiple auxiliary heads for object detection (e.g., critical objects and lanes), traffic state, and motion prediction of dynamic agents. The scene queries serve as tokens representing the key information of the current scene.

Additionally, we employ a set of history queries *Qh*∈ℝ*Nh*×*Cq* and a long-term memory bank *M*∈ℝ(*Nh*×*n*)×*Cq* to efficiently retrieve and store essential historical information (e.g., previous road conditions and ego status), where *Nh* is the number of history queries and *n* is the maximum history frame length. The process of analyzing *Qh* and *M* can be formulated as:



where *Pt* denotes the relative timestamp embedding.

Subsequently, the updated history queries  are stored in the memory bank *M* following the First-In-First-Out (FIFO) replacement policy, formulated as:



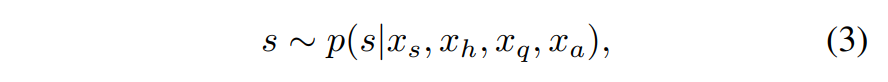
where *t* is the current frame time.

Finally, we utilize a two-layer MLP to convert the updated history queries  and current scene features *Qs* to history tokens *xh* and scene tokens *xs* in the reasoning space of LLM.

**(2) Large Language Model**

The user instruction is first encoded into language tokens *xq*∈ℝ*L*×*C* by the text tokenizer, where *L* is the token length and *C* is the dimension of LLM. Then, the scene tokens *xs* and history tokens *xh* are combined with the language tokens *xq* and fed into LLM.

Meanwhile, we design a planning QA template with a special planning token *s* for LLM as the final QA to accumulate the understanding and reasoning context of the entire driving scenario to the *s*, formally written as:



where *xa* denotes the generation answer of LLM. The embedding of the planning token *s* will serve as a condition to control the trajectory generation.

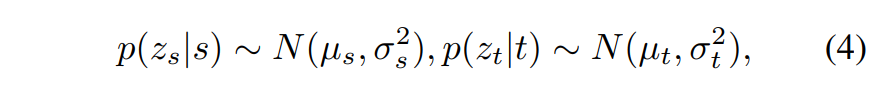
However, there is still a lack of high-quality VQA annotations within closed-loop simulation environments to train LLMs for comprehensively understanding driving scenarios. Thus, we extend the Bench2Drive dataset via a fully automatic VQA annotation pipeline powered by Qwen2- VL [57] and propose our VQA dataset, Chat-B2D, expecting to further promote the research of VLM on closed-loop simulation. We provide detailed information on Chat-B2D and its annotation pipeline in the Appendix.

**(3) Generative Planner**

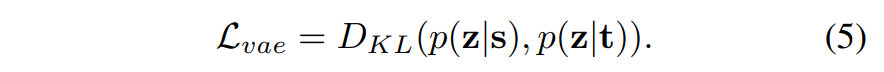
Generative models [14, 28, 48] can effectively capture intrinsic features within data by learning the distribution of the data.

Inspired by the generative domain, we introduce a generative planner to bridge the gap between the reasoning and action space. Specifically, we formulate the current trajectory *a* in action space as a conditional probability distribution *p*(*a*|*s*), where *s* is the planning token. To construct *p*(*a*|*s*), there are many excellent methods in the generation field (e.g., variational autoencoders (VAE) [28] and diffusion model [48]).

As there are essential differences in the distribution between the reasoning space of VLM and the action space of trajectory, **we use the VAE** [28] **model to align them in the Gaussian distribution.** We employ two-layer MLPs to project both the state *s* and the ground-truth trajectory *t* into Gaussian variables *z* in the latent space, denoted as:



where *N*(*µ*, *σ*2) denotes a Gaussian distribution with a mean of *µ*, and standard deviation of *σ*. We then use Kullback-Leibler divergence loss to enforce distribution matching, represented as:



Finally, we use the GRU decoder in GenAD [70] to decode the trajectory from the latent space *z*.

Significantly, the functions of VAE in this paper are not the same as VAE of GenAD. We only use a single token encoded in the reasoning space from the perspective of the ego vehicle as input, aiming to bridge the gap between reasoning space and action space.

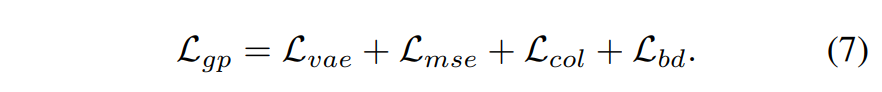
**Additionally, we also attempt to replace the VAE with alternative generative models, such as the diffusion model for trajectory generation. Benefiting from the proposed method that bridges the gap between the reasoning and action space through distribution learning in latent space, our framework still demonstrates superior performance compared to other methods (detailed in Sec. 4.5).**

**(4) Training Objectives**

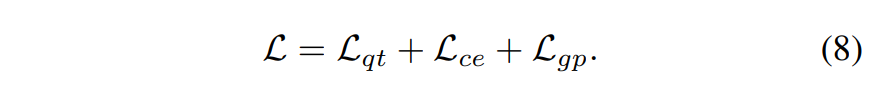
For the detection task of the proposed QT-Former, the detection loss is defined as *Ldet* = *Lcls* + *Lreg*, where *Lcls* is focal loss [34] and *Lreg* is L1 loss. For the traffic state and motion prediction, the losses are defined as *Ltra* and *Lm* = *Lmcls* + *Lmreg*, respectively, where *Ltra* and *Lmcls* are focal loss, and *Lmreg* is L1 loss. The total loss of QTFormer is:



For the LLM, we leverage the auto-regressive crossentropy loss *Lce*. For the generative planner in our framework, *Lvae* is the Kullback-Leibler divergence loss used to align the reasoning space and action space. Following VAD [25], we adopt the collision loss *Lcol*, boundary loss *Lbd*, and MSE loss *Lmse* for the planning prediction. The total loss of the generative planner is:



In summary, the total loss of the proposed ORION is:



**Dataset:**

We train and evaluate ORION on the Bench2drive dataset [23], a closed-loop evaluation protocol under CARLA V2 [11] for E2E autonomous driving. Additionally, we compare our method with other SOTA baselines on nuScenes [6] open-loop evaluation, which will be provided in the Appendix.

**Evaluation Metrics:**

Bench2drive includes five metrics for closed-loop evaluation: Driving Score (DS), Success Rate (SR), Efficiency, Comfortness, and Multi-Ability.

For open-loop evaluation, we use the L2 distance error and the collision rate.

Additionally, we use CIDEr [56], BLEU [41], and ROUGE-L [33] to evaluate the performance of ORION on VQA tasks.

关于这篇paper的更多解读还可参见如下网址：

<https://zhuanlan.zhihu.com/p/1890790479802638885>